



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA TELEMÁTICA

RECONOCIMIENTO DE EMOCIONES A TRAVÉS DE LA VOZ

Autor: Marta Castañeda Albillos

Director: Miguel Ángel Sanz Bobi

Madrid

Julio 2018

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Reconocimiento de emociones a través de la voz

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2017/18 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Marta Castañeda Albillos

Fecha: 18/07/2018

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Miguel Ángel Sanz Bobi

Fecha: 18/07/2018

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

La autora Dña. MARTA CASTAÑEDA ALBILLOS DECLARA ser el titular de los derechos de propiedad intelectual de la obra: RECONOCIMIENTO DE EMOCIONES A TRAVÉS DE LA VOZ, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún

- derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
 - c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
 - d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 18 de julio de 2018

ACEPTA



Fdo Marta Castañeda Albillos

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA TELEMÁTICA

RECONOCIMIENTO DE EMOCIONES A TRAVÉS DE LA VOZ

Autor: Marta Castañeda Albillos

Director: Miguel Ángel Sanz Bobi

Madrid

Julio 2018

Agradecimientos

En primer lugar, quiero agradecerle a mi director Miguel Ángel todo el tiempo dedicado en este proyecto, el cual fue difícil de encarrilar en un principio pero que poco a poco, gracias a sus aportaciones y el trabajo realizado, fue cogiendo forma.

Y también agradecer a mi familia todo su apoyo durante estos años, sin ellos no hubiera sido posible.

Muchas gracias.

RECONOCIMIENTO DE EMOCIONES A TRAVÉS DE LA VOZ

Autor: Castañeda Albillos, Marta.

Director: Sanz Bobi, Miguel Ángel.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En el presente proyecto se ha realizado un estudio comparativo de unas técnicas de clasificación de datos utilizando un conjunto de características de la señal de la voz con el fin de encontrar las propiedades de la voz más significativas a la hora de reconocer emociones y, a su vez, definir la técnica con la que se obtienen mejores resultados. Para ello, se han implementado tres soluciones: un open-software basado en máquinas de soporte vectorial (SVM), unos mapas autoorganizados (SOM) y los SOMs complementados con árboles de decisión.

Palabras clave: reconocimiento, técnica de clasificación, emoción, voz.

1. Introducción

Durante los últimos años, los sistemas que llevan integrado un reconocimiento del lenguaje natural humano han adquirido mucha importancia en la mejora de la interacción persona-máquina. Pero no solo es importante identificar qué dice una persona sino cómo lo está transmitiendo. De esta manera, las emociones poseen un papel muy significativo, ya que las propiedades de la voz varían según cómo se encuentre el hablante. Por lo tanto, el reconocimiento de emociones a través de la voz proporciona información muy útil en diversas aplicaciones en ámbitos como el de la sanidad, la seguridad y la atención al cliente.

Para el reconocimiento del que se viene hablando, actualmente existen múltiples productos en el mercado, pero con casos de uso muy concretos. Las técnicas de reconocimiento de patrones, tras la extracción de características de la señal de la voz, nos pueden ayudar a identificar emociones realizando un buen entrenamiento. Estos métodos se pueden dividir en supervisados o no supervisados, y pueden basarse en múltiples algoritmos como redes neuronales artificiales, análisis vectorial, etc. (ver [Estado del arte 1.2](#)).

2. Definición del Proyecto

Este proyecto se propuso con el fin de identificar las propiedades de la voz más significativas para el reconocimiento de emociones y, a su vez, realizar una comparativa de los resultados que se obtienen al utilizar diferentes técnicas de clasificación de datos para ello. (ver [Objetivos 3.2](#))

Es interesante poder identificar la técnica de reconocimiento de patrones que realice una mejor clasificación de las emociones a través de la voz para la comercialización del proyecto en ámbitos como la sanidad, la seguridad, la atención a los usuarios, así como en el hogar y automóvil (ver [Justificación 3.1](#)).

El proyecto se ha realizado, basándose en una metodología de desarrollo de software tradicional, en seis fases (ver [Metodología 3.3](#)): trabajos previos, especificación de requisitos, las tres soluciones basadas en diferentes técnicas de clasificación y, por último, conclusiones. A su vez, las fases correspondientes a cada solución se han dividido en dos subfases:

- I. En la fase de entrenamiento, se han utilizado un conjunto de muestras por cada emoción que se quería reconocer, que en este proyecto han sido *enfado*, *asco*, *tristeza*, *sorpresa* y *neutral*, para que cada técnica aprenda los patrones que posee la voz en las diferentes emociones.
- II. En la fase de reconocimiento, en cada una de las soluciones se han introducido muestras completamente nuevas con el fin de verificar que dicho método realiza una buena clasificación de los datos.

La planificación que se ha seguido para realizar este proyecto se muestra en la *Ilustración 1* (ver [Planificación 3.4.1](#)):

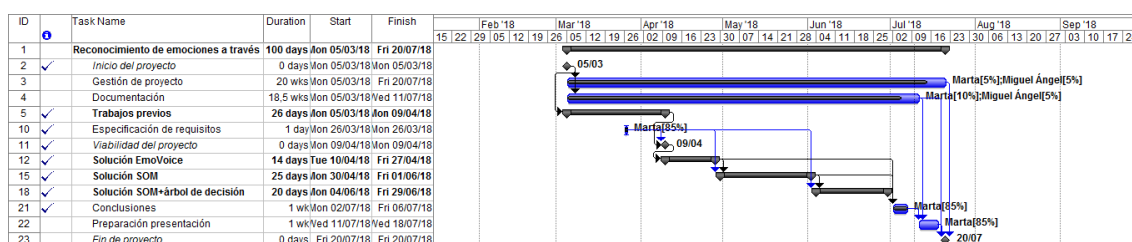


Ilustración 1. Planificación – resumen

El coste de realizar este proyecto, teniendo en cuenta únicamente en coste del personal, se resume en un total de 11.420 € (ver [Estimación económica 3.4.2](#)).

3. Descripción del modelo/sistema/herramienta

Para la realización de este proyecto, se utiliza un sistema de extracción de características de la señal de la voz que se integra en tres soluciones basadas en diferentes métodos de reconocimiento de patrones. Estas herramientas se definen a continuación:

- A. Para extraer las características de la voz se ha utilizado la herramienta *openSMILE* [1], la cual, a través de un archivo de configuración, se define el tratamiento que se quiere realizar a un audio y las características que se quieren extraer del mismo.
- B. La herramienta del punto A se ha integrado en tres soluciones basadas en diferentes técnicas de reconocimiento de patrones:
 1. Un open-software basado en una máquina de soporte vectorial (SVM).
 2. Unos mapas autoorganizados (SOM).
 3. Y, por último, los SOMs junto a árboles de decisión.

La solución del punto 1 consiste en un producto open-sw, llamado EmoVoice [2], que integra la herramienta *openSMILE* y utiliza la máquina de soporte vectorial (SVM) como método de clasificación de datos (ver [Solución EmoVoice 5](#)). En la *Ilustración 2* se muestra la arquitectura de este sistema.

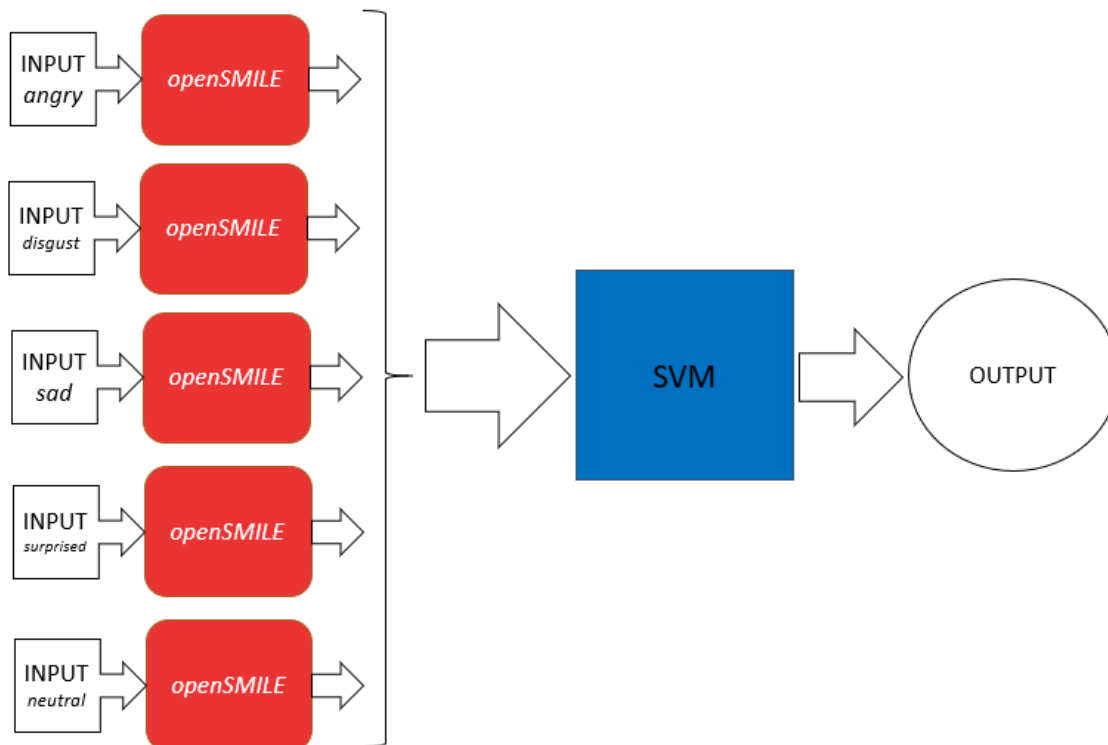


Ilustración 2. Arquitectura solución EmoVoice

La solución de los mapas autoorganizados (SOM) [3] consiste en el diseño e implementación de cinco mapas autoorganizados, uno por cada emoción, los cuales memorizan o aprenden los patrones que posee la voz cuando expresa cada una de esas emociones. Como se ha mencionado, esta solución integra la herramienta *openSMILE* para la extracción de las características de la voz, previa al entrenamiento de los mapas (ver [Solución mapa autoorganizado 6](#)). En la *Ilustración 3* se puede observar que la arquitectura de cada SOM está formada por una capa de 9 neuronas artificiales tomando como entrada las características de la voz extraídas de *openSMILE*.

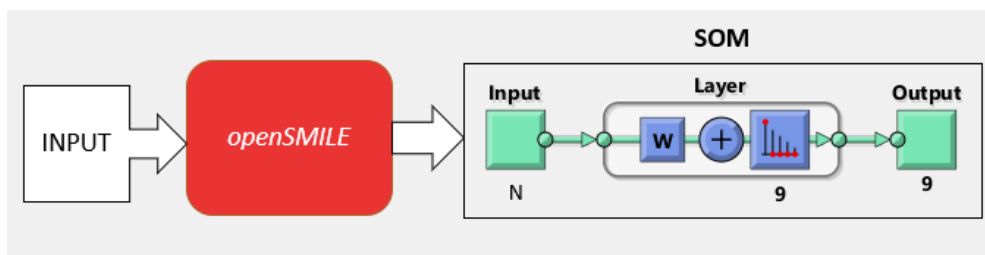


Ilustración 3. Arquitectura SOM

Por último, la solución de los mapas autoorganizados junto a árboles de decisión [4] trata de refinar los resultados de la solución anterior integrando la herramienta WEKA para la extracción de los árboles de decisión (ver [Solución SOM junto a árbol de decisión 7](#)). La arquitectura completa de esta solución se muestra en la *Ilustración 4*.

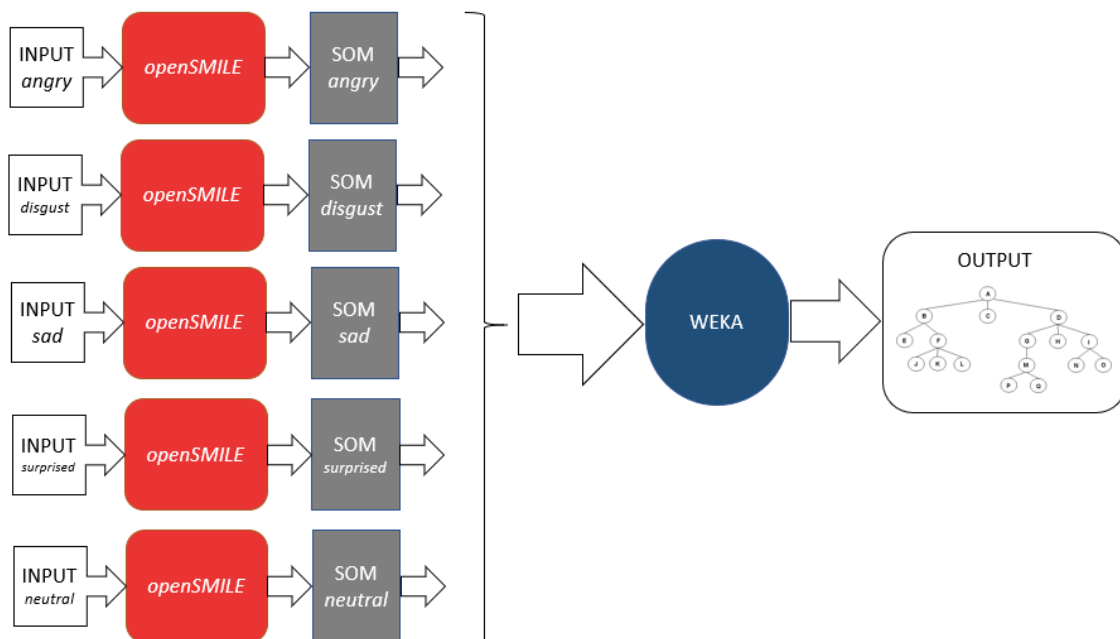


Ilustración 4. Arquitectura solución SOM junto a árbol de decisión

4. Resultados

Se han realizado pruebas en las tres soluciones, generando diferentes resultados en cada una de esas pruebas:

1. Tras realizar pruebas con la solución EmoVoice, se han obtenido resultados con muy buena fiabilidad en la clasificación de cada emoción con un porcentaje del 83% de aciertos (ver [Análisis de pruebas 5.3](#)).
2. Con la solución de los mapas autoorganizados, tras las pruebas, se ha obtenido un porcentaje de aciertos del 10% (ver [Análisis de pruebas 6.3](#)).
3. Y, con la integración de los árboles de decisión a la solución de los mapas autoorganizados, se mejoran los resultados obteniendo una media del 35% de aciertos (ver [Análisis de pruebas 7.3](#)).

5. Conclusiones

Se van a separar las conclusiones según las soluciones empleadas para alcanzar los objetivos de este proyecto (ver [Conclusiones y trabajos futuros 8](#)):

1. Con las pruebas realizadas en el sistema de EmoVoice, se puede observar que es con la solución que mejores resultados se obtienen, pero no nos ayuda a la identificación de las características de la voz.
2. Por otro lado, se desconoce si los malos resultados de la solución de los mapas autoorganizados proceden del tamaño de los datos utilizados para el entrenamiento o por el número de neuronas utilizados en el proceso. Se quedaría como trabajo futuro ampliar pruebas en el ámbito de las redes neuronales artificiales.
3. Y, para finalizar, se ha observado que al complementar la solución de los SOMs con árboles de decisión se mejoran los resultados significativamente.

6. Referencias

- [1] <https://audeering.com/technology/opensmile/>
- [2] <https://rawgit.com/hcmlab/emovoice/master/docs/index.html>
- [3] https://es.wikipedia.org/wiki/Mapa_autoorganizado
- [4] <https://www.tesisenred.net/bitstream/handle/10803/9441/tjjmm1de1.pdf?sequence=1>

RECOGNITION OF EMOTIONS THROUGH VOICE

Author: Castañeda Albillos, Marta.

Supervisor: Sanz Bobi, Miguel Ángel.

Collaborating Entity: ICAI – Universidad Pontificia Comillas.

ABSTRACT

In this project, a comparative study of data classification techniques using a set of characteristics of the voice signal has been carried out. The main objective has been to find the most significant properties of the voice when recognizing emotions and identify the technique with which best results are obtained. Therefore, three solutions have been implemented: an open-software based on vector support machines (SVM), self-organized maps (SOM) and SOMs complemented with decision trees.

Keywords: recognition, classification technique, emotion, voice.

1. Introduction

During the last years, systems that have integrated an acknowledgment of human natural language have acquired great importance in the improvement of the human-machine interaction. But it is not only important to identify what a person says but how he / she is transmitting the message. So, emotions play a very significant role because the properties of the voice vary according to how the speaker is. Therefore, the recognition of emotions through voice provides very useful information in various applications in areas such as health, safety and customer service.

For the recognition that has been discussed, there are currently multiple products on the market, but with very specific use cases. The techniques of pattern recognition, after the extraction of characteristics of the voice signal, can help us identify emotions by performing a good training. These methods can be divided into supervised or unsupervised learning models and can be based on multiple algorithms such as artificial neural networks, vector analysis and so on (see [Estado del arte 1.2](#)).

2. Definition of the project

This project was proposed to identify the most significant properties of the voice for the recognition of emotions and to make a comparison of the results obtained by using different data classification techniques (see [Objetivos 3.2](#))

The point resides on the ability of identifying the technique of pattern recognition that performs a better classification of emotions through the voice to implement the project in areas such as health, safety and customer service. (see [Justificación 3.1](#)).

The project has been carried out in six phases, based on a traditional software development methodology (see [Metodología 3.3](#)): previous work, specification of requirements, the three solutions based on different classification techniques, and finally, conclusions. At the same time, the phases of each solution have been divided into two sub-phases:

- I. In the training phase, a set of samples was used for each emotion that was wanted to be recognized (in this project have been *angry*, *disgust*, *sad*, *surprised* and *neutral*), so that each technique learns the patterns of the voice in different emotions.
- II. In the recognition phase, completely new samples have been introduced in each of the solutions to verify that the method performs a good classification of the data.

The planning that has been followed to carry out this project is shown in Figure 1 (see [Planificación 3.4.1](#)):

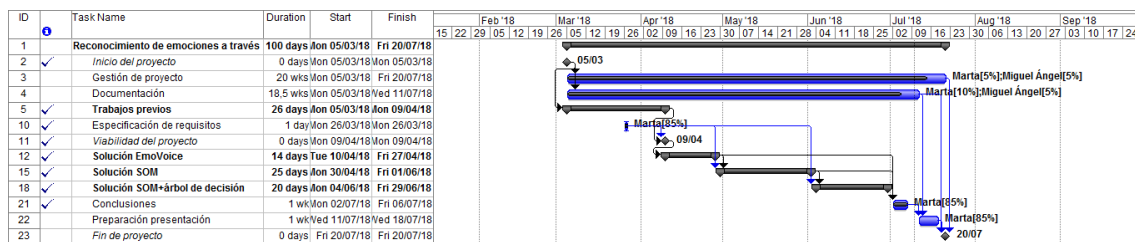


Figure 1. Planning

The cost of carrying out this project is summarized in a total of 11,420 € - considering only the cost of the staff (see [Estimación económica 3.4.2](#))

3. Description of the model/system/tool

For the realization of this project, a system that extracts the characteristics of the voice is integrated into three solutions based on different pattern recognition methods. These tools are defined below:

- A. To extract the characteristics of the voice, we have used the *openSMILE* tool, which, through a configuration file, defines the treatment to be carried out on an audio and the characteristics that we want to extract from it.
- B. The point A tool has been integrated into three solutions based on different pattern recognition techniques:
 1. An open-software based on a vector support machine (SVM).
 2. Self-organized maps (SOM).
 3. And finally, the SOMs next to decision trees.

The solution of point 1 consists of an open-sw product called EmoVoice, which integrates the openSMILE tool and uses the vector support machine (SVM) as a data classification method (see [Solución EmoVoice 5](#)). In Figure 2 the architecture of this system is shown.

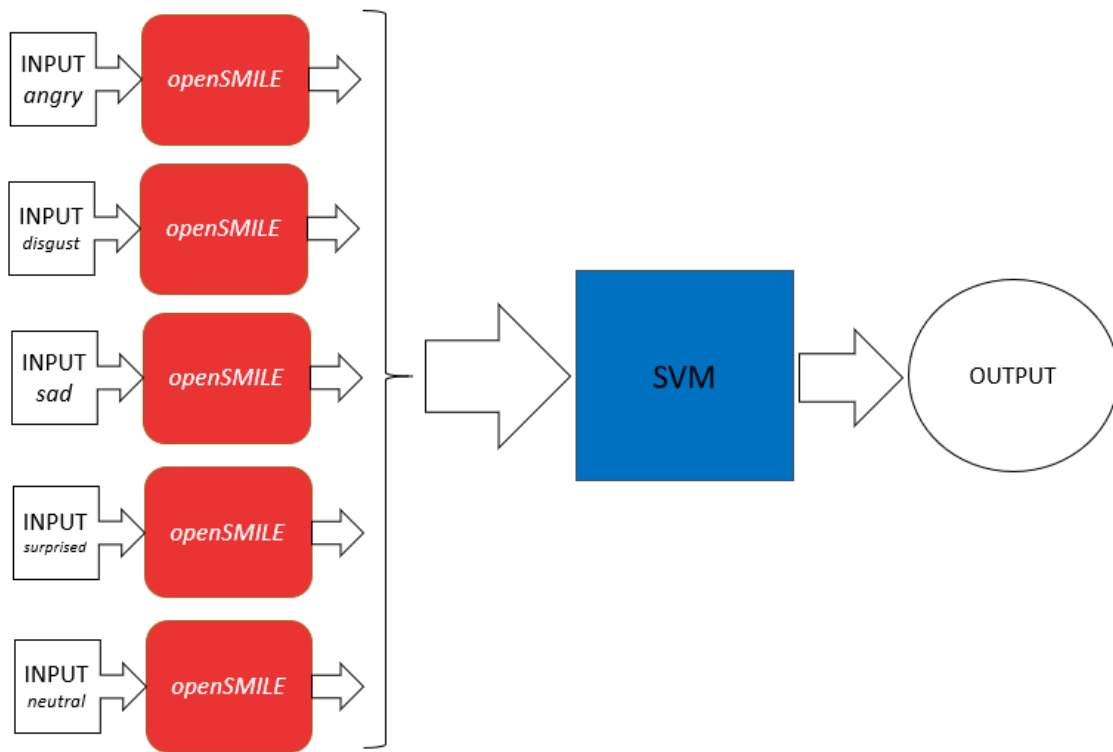


Figure 2. EmoVoice architecture

The solution of self-organized maps (SOM) consists on the design and implementation of five self-organized maps, one for each emotion, which memorize or learn the patterns of the voice when it expresses each one of those emotions. As mentioned, this solution integrates the *openSMILE* tool for extracting the characteristics of the voice, prior to maps training (see [Solución mapa autoorganizado 6](#)). In Figure 3, the architecture of each SOM has a layer of 9 artificial neurons taking as input the characteristics of the voice extracted from *openSMILE*.

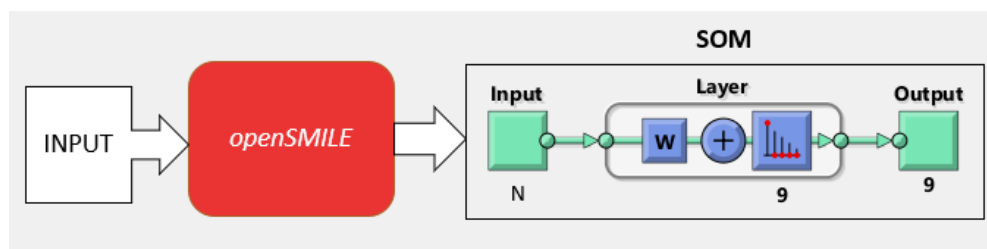


Figure 3. SOM architecture

Finally, the solution of self-organized maps with decision trees tries to refine the results of

the previous solution by integrating the WEKA tool for the extraction of the decision trees (see [Solución SOM junto a árbol de decisión 7](#)). The complete architecture of this solution is shown in Figure 4.

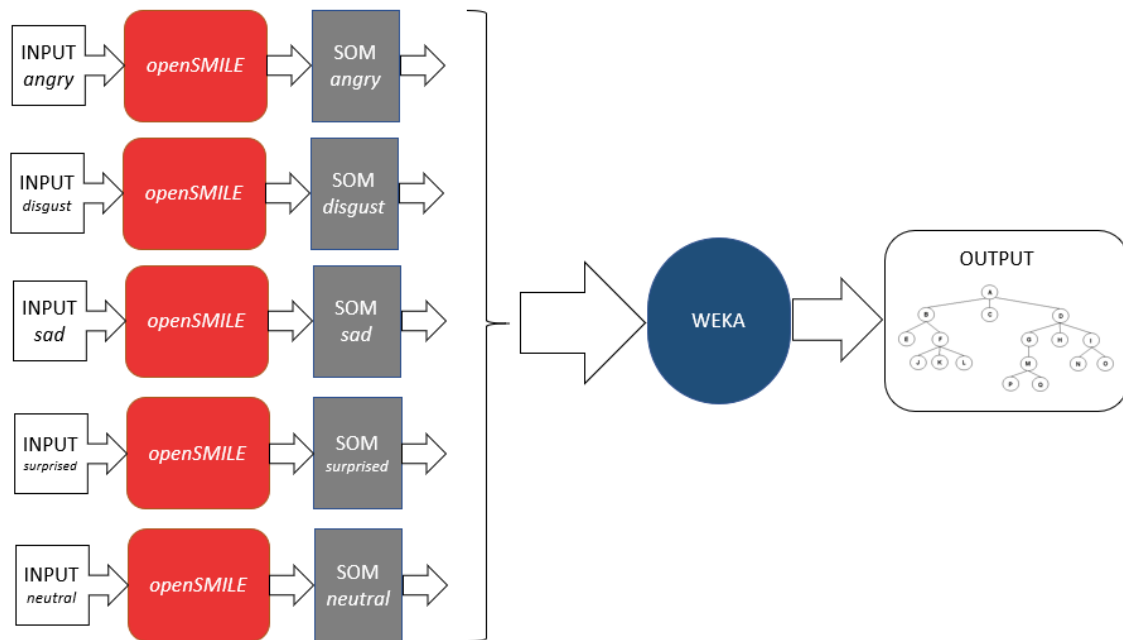


Figure 4. SOM with decision trees architecture

4. Results

Tests have been carried out in the three solutions, generating different results in each of these tests:

1. After testing with the EmoVoice solution, results have been obtained with high reliability in the classification of each emotion with a percentage of 83% (see [Análisis de pruebas 5.3](#)).
2. With the solution of the self-organized maps, after the tests, a percentage of correct answers of 10% has been obtained (see [Análisis de pruebas 6.3](#)).
3. And, with the integration of the decision trees to the solution of the self-organized maps, the results improved obtaining an average of 35% of correct classifications (see [Análisis de pruebas 7.3](#)).

5. Conclusions

The conclusions will be separated according to the solutions used to achieve the objectives of this project (see [Conclusiones y trabajos futuros 8](#)):

1. With the tests carried out in the EmoVoice system, this system is the solution that better results obtain, but it does not help us to identify the characteristics of the voice.
2. On the other hand, the reason for poor results in the self-organized maps is unknown. It could be because of the size of the data used for training or the number of neurons

used in the process. A future work would be to expand the tests in the field of neural networks.

3. And, finally, it has been observed that by complementing the solution of the SOMs with decision trees, the results are significantly improved.

6. References

- [1] <https://audeering.com/technology/opensmile/>
- [2] <https://rawgit.com/hcmlab/emovoice/master/docs/index.html>
- [3] https://es.wikipedia.org/wiki/Mapa_autoorganizado
- [4] <https://www.tesisenred.net/bitstream/handle/10803/9441/tjjmm1de1.pdf?sequence=1>

Índice de la memoria

Capítulo 1. Introducción	7
1.1 Motivación del proyecto.....	8
1.2 Estado del arte	8
1.2.1 Características de la voz	10
1.2.2 Técnicas de clasificación de datos o reconocimiento de patrones.....	11
Capítulo 2. Descripción de las Tecnologías.....	14
2.1 openSMILE	14
2.2 EmoVoice.....	15
2.3 Mapa Autoorganizado (SOM).....	16
2.4 Árbol de decisión.....	17
Capítulo 3. Definición del Trabajo	19
3.1 Justificación.....	19
3.1.1 Call Centers.....	19
3.1.2 Seguridad.....	20
3.1.3 Hogar y Automóvil	21
3.1.4 Sanidad.....	21
3.2 Objetivos	22
3.3 Metodología.....	22
3.3.1 Fase trabajo previo	23
3.3.2 Fases de cada solución.....	25

ÍNDICE DE LA MEMORIA

3.4 Planificación y Estimación Económica	25
3.4.1 Planificación.....	25
3.4.2 Estimación económica.....	27
Capítulo 4. Requisitos del proyecto	29
4.1 Requisitos generales	29
4.2 Requisitos de interfaz	30
4.3 Requisitos de hardware	31
Capítulo 5. Solución EmoVoice	32
5.1 Análisis del Sistema	32
5.2 Implementación.....	33
5.2.1 Preparación de los datos.....	33
5.2.2 Extracción de características.....	34
5.2.3 Fase de entrenamiento	35
5.2.4 Fase de reconocimiento.....	36
5.3 Análisis de pruebas.....	37
Capítulo 6. Solución mapa autoorganizado (SOM)	39
6.1 Análisis del sistema	39
6.2 Implementación SOMs.....	40
6.2.1 Codificación	40
6.2.2 Fase de entrenamiento	42
6.2.3 Fase de reconocimiento.....	46
6.3 Análisis de pruebas.....	48

Capítulo 7. Solución SOM junto a árbol de decisión.....	52
7.1 Análisis del sistema.....	52
7.2 Implementación.....	54
7.3 Análisis de pruebas.....	55
7.3.1 Fase refinamiento SOMs.....	55
7.3.2 Fase reducción de datos.....	60
Capítulo 8. Conclusiones y Trabajos Futuros.....	66
Capítulo 9. Bibliografía.....	68
ANEXO A: Archivo de configuración compare.conf.....	70
ANEXO B: Características openSMILE.....	93
ANEXO C: SOMs características bloque1_1.....	102
ANEXO D: Ejemplo archivo ARFF.....	105
ANEXO E: Tabla fiabilidad y error.....	112
Bloque 1.....	112
Bloque 2.....	115
Bloque 3.....	115
Bloque 4.....	116
Bloque 5.....	117

Índice de figuras

Ilustración 1. Planificación – resumen	11
Ilustración 2. Arquitectura solución EmoVoice	12
Ilustración 3. Arquitectura SOM	13
Ilustración 4. Arquitectura solución SOM junto a árbol de decisión	13
Ilustración 5. Arquitectura EmoVoice	33
Ilustración 6. Procesamiento de la señal de audio a través de EmoVoice	37
Ilustración 7. Interfaz EmoVoice	37
Ilustración 8. Arquitectura SOM	40
Ilustración 9. SOMs características bloque1_1.	43
Ilustración 10. SOM Angry Neurona 9.	44
Ilustración 11. SOM Disgust Neurona 4.	44
Ilustración 12. SOM Sad Neurona 1.	44
Ilustración 13. SOM Surprised Neurona 4.	45
Ilustración 14. SOM Neutral Neurona 1.	45
Ilustración 15. Arquitectura sistema completo	53
Ilustración 16. Árbol de decisión bloque1_1	56

Ilustración 17. Árbol de decisión fase reducción de datos 61

Índice de tablas

Tabla 1. Matriz de confusión pruebas EmoVoice.	38
Tabla 2. Matriz Bloque 1_1.....	49
Tabla 3. Matriz Bloque1_2.....	50
Tabla 4. Matriz Bloque1_3.....	50
Tabla 5. Características con árbol de decisión fiabilidad > 90 %	61
Tabla 6. Propiedades de la voz resultantes SOM con Árbol de decisión fase 2.....	62
Tabla 7. Características openSMILE organizadas en bloques.	101
Tabla 8. Tabla fiabilidad/error bloque 1	115
Tabla 9. Tabla fiabilidad/error bloque 2.....	115
Tabla 10. Tabla fiabilidad/error bloque 3.....	115
Tabla 11. Tabla fiabilidad/error bloque 4.....	117
Tabla 12. Tabla fiabilidad/error bloque 5.....	119

Capítulo 1. INTRODUCCIÓN

Durante los últimos años se ha producido una gran evolución en la cantidad de dispositivos que manejamos las personas y los servicios que utilizamos en los mismos. Recientemente se puede ver cómo la voz cobra importancia en dicha interacción, intentando que los dispositivos puedan reconocer un lenguaje natural humano y actuar en consonancia. Este proyecto se propuso con el fin de avanzar en la identificación de las emociones que expresa una persona al hablar, realizando una comparativa de la efectividad real de técnicas existentes para la identificación de las características de la señal de la voz más importantes para esta tarea.

Para comenzar se ha realizado un estudio sobre el estado del arte de esta cuestión, definiendo los principales parámetros identificados hoy en día para el proceso de reconocimiento de emociones, así como las técnicas de clasificación más utilizadas.

Tras realizar ese estudio, hubo un proceso de familiarización con una herramienta de extracción de parámetros de la voz que se encuentra actualmente en el mercado llamada *openSMILE*, la cual se detallará más adelante.

El paso siguiente consiste en una serie de pruebas que se realizaron con diferentes soluciones o técnicas de clasificación, concretamente:

1. un open-software basado en una máquina de soporte vectorial (SVM),
2. uso de mapas autoorganizados (SOM),
3. y una solución que integra un árbol de decisión sobre un mapa autoorganizado.

INTRODUCCIÓN

Para llevar a cabo todas estas pruebas fueron necesarias una serie de muestras de audios que mostrarán diversas emociones. Por lo tanto, un trabajo previo a esas pruebas fue la búsqueda de una base de datos de audios que mostrarán distintos estados de ánimo.

Tras el proceso de pruebas, se muestra en este documento una serie de conclusiones que muestran las características y las técnicas de clasificación más útiles para el reconocimiento de las emociones.

1.1 MOTIVACIÓN DEL PROYECTO

Hoy en día, mejorar la interacción entre las personas y las máquinas parece casi una necesidad más en una sociedad que está cada vez más *tecnologizada*. Dicha interacción puede ser, por ejemplo, el servicio telefónico de atención al cliente de una empresa. En este caso, dicha comunicación tiene que ser lo más satisfactoria posible.

En la mayoría de las ocasiones, la forma en que una persona transmite un mensaje puede cambiar completamente su significado. De esta manera, entran en juego las emociones ya que éstas nos condicionan a la hora de transmitir un mensaje. Por ello, la automatización del reconocimiento de emociones a través de la voz humana puede suponer un gran avance en este aspecto, proporcionando una información muy útil.

1.2 ESTADO DEL ARTE

Antes de conocer cuál es el estado de la cuestión del proceso de reconocimiento de las emociones a través de la voz [1], es necesario conocer cuáles son las propiedades de la voz que nos proporcionan la información para saber qué estado de ánimo posee una persona.

Las emociones son reacciones psicofisiológicas que representan modos de adaptación a ciertos estímulos del individuo cuando percibe cosas a su alrededor. Cuando una persona

INTRODUCCIÓN

experimenta una emoción, surgen una serie de manifestaciones físicas que modifican la flexibilidad de la musculatura impactando en la respiración de la persona. Cada vez que nuestra respiración cambia, se modifican las cualidades de la voz, afectando en la forma en que una persona transmite una serie de palabras, sobre todo, desde el punto de vista fonético [2]. Por ello, parametrizando la señal de la voz podemos llegar al reconocimiento que estamos tratando.

Las emociones se diferencian en emociones primarias y secundarias, así como en positivas y negativas. Este proyecto se va a enfocar en el reconocimiento de, sobre todo, las emociones primarias, como base de futuras aplicaciones.

Las principales características de la señal de la voz para reconocer emociones son la frecuencia fundamental junto a su contorno, la tasa de la voz y la forma en que se distribuye la energía en el espectro de voz.

La evolución tecnológica ha permitido la implantación de una serie de algoritmos de reconocimiento de emociones. De esta manera, una vez extraídas las características de la voz podemos automatizar dicho reconocimiento mediante el entrenamiento de un modelo de clasificación. Hoy en día, existen muchos modelos de reconocimiento de patrones, que pueden llevarse a cabo a través de métodos de clasificación supervisada o no supervisada.

De hecho, existen productos en el mercado que cubren el objetivo de reconocimiento de emociones a través de la voz [3] [4] [5]. Un ejemplo de ello es el producto Verbio ASER (Automatic Speech Emotional Recognition), lanzado por la multinacional Verbio Technologies. Esta tecnología es un reconocedor de voz que integra la detección emocional, que fue desarrollado enfocado a la atención del usuario. Otro ejemplo son las aplicaciones móviles de *Moodies* [6], lanzado por la compañía israelí Beyond Verbal, trata de un software que capta la emoción que tiene una persona mientras habla por teléfono. Esta herramienta puede ser útil tanto a nivel personal como para que una empresa pueda mejorar su servicio

INTRODUCCIÓN

telefónico de atención al cliente. De los productos mencionados, no existe información sobre las tecnologías utilizadas, por lo que no se puede aprovechar para aplicarlo o integrarlo en otras soluciones sobre otros casos de uso (ver [3.1 Justificación](#)).

En este capítulo se mencionarán algunas de las características de la voz que son más significativas para el reconocimiento, así como las técnicas de clasificación más utilizadas para ello actualmente.

1.2.1 CARACTERÍSTICAS DE LA VOZ

En este apartado se explica detalladamente qué es la señal de la voz haciendo hincapié en las características más importantes *a priori* de la voz en función de la emoción que está expresando el hablante [7].

La señal de la voz humana está formada por un conjunto de sonidos generados por el aparato fonador. Esta señal como cualquier otra se puede transformar a través de un micrófono en señal eléctrica, pudiendo ser tratada para la extracción de sus propiedades.

La señal de la voz puede ser representada tanto en el dominio del tiempo como en el de la frecuencia. La representación de esta señal en el tiempo puede aportar información sobre su energía, así como la representación en el dominio de la frecuencia proporciona las características espectrales del audio en cuestión.

Como el cambio de las características se produce de una manera muy lenta en la señal de la voz, es necesario utilizar una ventana con el tamaño justo para que no elimine las propiedades más importantes para el reconocimiento y a su vez las propiedades que nos interesan no cambien. La función de la ventana es la siguiente:

$$f_s(n; m) = s(n) * w(m - n)$$

Siendo $s(n)$ es la señal de la voz y $w(m - n)$ la ventana.

A continuación, se detallan algunas de las características de la voz que aportan información a la hora de reconocer la emoción del hablante.

1.2.1.1 Frecuencia fundamental o pitch

El pitch o frecuencia fundamental (F0) es la frecuencia a la que vibran las cuerdas vocales. Una persona, dependiendo del estado de ánimo que tenga, variará la tensión de las cuerdas vocales, así como la presión del aire que procede de la cavidad pulmonar.

1.2.1.2 Energía

La energía de la señal de la voz tiene la siguiente forma:

$$E_s(m) = \frac{1}{N_w} \sum_{n=m-N_w-1}^m |f_s(n; m)|^2$$

Siendo N_w la duración de la energía.

Esta energía varía según la excitación del hablante, lo cual puede aportar información a la hora de determinar la emoción del hablante.

1.2.2 TÉCNICAS DE CLASIFICACIÓN DE DATOS O RECONOCIMIENTO DE PATRONES

Las técnicas de clasificación de datos [8] tienen como objetivo generalizar una serie de comportamientos a partir del suministro de unos ejemplos, de tal manera que la máquina *aprenda* cuáles son los patrones representativos dependiendo de la muestra de entrada que se le introduzca.

INTRODUCCIÓN

Hoy en día existen muchas técnicas capaces de realizar este proceso, estas se dividen en métodos de clasificación supervisados y no supervisados.

1.2.2.1 Métodos de clasificación supervisados

El aprendizaje supervisado consiste en una técnica que genera una función a partir de unos datos de entrenamiento. Estos datos suelen ser vectores formados por los datos de entrada y los resultados deseados a partir de dichas entradas. De esta forma, este método genera una función capaz de predecir el resultado de una muestra completamente nueva a partir del entrenamiento realizado previamente.

1.2.2.1.1 Redes neuronales artificiales

Una red neuronal artificial (RNA) [9] [10] es un modelo computacional basado en un conjunto de neuronas artificiales, basadas en las redes neuronales biológicas. Estas redes están formadas por neuronas organizadas en capas que se conectan entre sí a través de unos enlaces que tienen asociados un peso. A través del entrenamiento, las redes neuronales crean su propia representación interna del problema pudiendo realizar predicciones sobre muestras que no habían sido expuestas anteriormente.

Típicamente las RNA se consideran métodos de aprendizaje supervisado, pero existen determinados tipos que tienen un aprendizaje no supervisado, como, por ejemplo, el mapa autoorganizado (SOM).

1.2.2.1.2 Máquina de soporte vectorial (SVM)

Las máquinas de soporte vectorial o SVMs [11] [12] son un conjunto de algoritmos de aprendizaje supervisado relacionados con problemas de clasificación y regresión. Los datos del entrenamiento [13] [14], previamente etiquetados en clases, son mapeados por medio de

INTRODUCCIÓN

funciones *Kernel* a un espacio de características de mayor dimensión. De esta forma, se puede predecir la clase a la que pertenece una nueva muestra.

1.2.2.1.3 Árboles de decisión

Un árbol de decisión [15] se utiliza en minería de datos para crear modelos predictivos automatizados. Estos árboles utilizan una serie de reglas para la predicción, por lo tanto, esta técnica se puede utilizar en el aprendizaje automático, entrenándolo para que identifique un valor o valores determinados. Existen diversos algoritmos, y aquí se va a utilizar C4.5, que es de los más relevantes en este ámbito. Esta técnica se detalla en el [apartado 2.4](#) del presente documento.

1.2.2.2 Métodos de clasificación no supervisados

Por otro lado, los métodos de clasificación no supervisados se diferencian del supervisado en que en este método no incluye un conocimiento *a priori*, es decir, construye un modelo de densidad con los datos de entrada analizando sus características, regularidades, correlaciones y categorías.

1.2.2.2.1 Mapas autoorganizados (SOM)

Los mapas autoorganizados o no supervisados [16] están basados en redes neuronales artificiales, y son generados a través del entrenamiento del mismo. Este tipo de red posee un aprendizaje no supervisado competitivo en el que las neuronas se autoorganizan según los datos introducidos en el mapa. Este método se explica más detenidamente en el [apartado 2.3](#).

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este capítulo se van a explicar las tecnologías que se van a mencionar a lo largo de este documento.

Como el objetivo principal es el reconocimiento de emociones a través de la voz se necesitará un analizador de emociones ([2.1](#)) así como herramientas que sean capaces de automatizar el proceso ([2.2](#), [2.3](#) y [2.4](#)).

2.1 *OPENS MILE*

openSMILE [17] es una herramienta open-source escrita en C++ que permite extraer características de una señal de audio. *SMILE* procede de *Speech & Music Interpretation by Large-space Extraction*. Este software permite interconectar libremente los componentes del que dispone para crear nuevas funciones personalizadas creando un archivo de configuración. Dicho archivo de configuración indica a *openSMILE* cómo se va a tratar el audio y qué características se le van a extraer.

OpenSMILE es multiplataforma y es capaz de realizar un procesamiento incremental rápido y eficiente en tiempo real. Sólo permite el tratamiento de archivos PCM WAVE (.WAV).

Entre las funciones de la señal de audio que puede realizar este software se encuentran funciones de ventana (Hamming, Hann, Gauss, Sine, ...), transformación rápida de Fourier, filtro de preacentuación, bancos de filtros FIR, autocorrelación y cepstrum.

A su vez, entre las características que es capaz de extraer son la energía de la señal, el volumen, los coeficientes MFCC, el tono, la calidad de la voz (Jitter, Shimmer) y los descriptores espectrales.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

También puede extraer funciones estadísticas de los audios como los medios, extremos, segmentos, picos, regresión, percentiles y los coeficientes DCT.

Esta herramienta está pensada para ser utilizada como parte del núcleo en un sistema y nunca como interfaz.

2.2 *EMOVOICE*

EmoVoice [18] es un conjunto de herramientas que permite entrenar y evaluar los modelos de clasificación principalmente en sistemas Windows de tal forma que complementa la herramienta *openSMILE* descrita anteriormente. De hecho, *EmoVoice* no solo complementa *openSMILE* sino que la integra en su sistema formando uno más complejo y sofisticado. Así, *EmoVoice* ofrece extracción de características, construcción y prueba de clasificadores, pudiendo incluso realizar un reconocimiento en línea.

Este software utiliza el clasificador *LibSVM* basado en Máquinas de Soporte Vectorial (SVM) pero también permite utilizar cualquier otro modelo de clasificación en su lugar.

La señal de audio requerida para el entrenamiento son audios con una frecuencia de muestreo de al menos 8 kHz y monomodo.

EmoVoice sigue una serie de pasos para poder predecir la emoción que expresan fragmentos de voz que se analizarán más adelante. Estos pasos se explicarán aplicados al proyecto en el [Capítulo 5](#).

2.3 MAPA AUTOORGANIZADO (SOM)

Un mapa autoorganizado o SOM [19] es un tipo de red neuronal artificial que en lugar de aplicar métodos de corrección de errores fomenta un aprendizaje basado en competición. Su entrenamiento se realiza usando aprendizaje no supervisado para producir una representación discreta del espacio de las muestras de entrada, llamado mapa. Al tratarse de un aprendizaje no supervisado o autoorganizado, durante el proceso de entrenamiento la red debe descubrir por sí misma rasgos comunes, correlaciones o categorías entre los datos de entrada. El SOM utiliza una función de vecindad para preservar las propiedades topológicas del espacio de entrada.

Los mapas autoorganizados [20] están formados por una serie de nodos o neuronas de entrada y una capa de neuronas que entran en competición. Estas neuronas tienen asociadas un vector de pesos -inicialmente aleatorios- desde la capa de neuronas de entrada, definiendo así su posición en el mapa. Los vectores de pesos poseen la misma dimensión que el vector de entrada. Estos pesos indican la importancia de la entrada dentro de la función de agregación de la neurona.

Los SOMs mapean un espacio de mayor dimensión a otro de menor dimensión.

Como en la mayoría de redes neuronales artificiales, los SOMs tienen un mecanismo de entrenamiento y otro para clasificación de datos. El entrenamiento consiste en construir el mapa utilizando una serie de muestras, mientras que el mapeo se encarga de clasificar una entrada nueva.

El algoritmo de clasificación de datos en un mapa autoorganizado consiste en los siguientes pasos:

1. Hacer un mapa de neuronas con vectores de pesos aleatorios.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

2. Tomar un vector de entrada $D(t)$.
 - a. Iterar por cada neurona del mapa.
 - i. Calcular la distancia entre el vector de entrada y los vectores de pesos de las neuronas que entran en competición del mapa.
 - ii. Mantener la neurona que ha tenido la menor distancia, esta neurona será el BMU (unidad de mejor correspondencia).
 - b. Actualizar las neuronas en la vecindad del BMU.
$$W_v(s+1) = W_v + \Theta(u, v, s)\alpha(s)(D(t) - W_v(s))$$
3. Incrementar s y volver al paso 2, mientras $s < \lambda$.

Siendo s la iteración actual, λ la cantidad total de iteraciones, t el índice del vector de entrada en el conjunto de datos de entrada D , v es el índice de una neurona en el mapa, W_v es el vector de pesos de la neurona v , u es el índice del BMU en el mapa, $\Theta(u, v, s)$ es la función de vecindad y, por último, $\alpha(s)$ es un restrictor de aprendizaje debido al progreso de las iteraciones.

2.4 ÁRBOL DE DECISIÓN

Un árbol de decisión [21] [22] es una técnica de clasificación supervisada que nos permite crear modelos predictivos automatizados que pueden emplearse, entre otras cosas, para el aprendizaje automático. La creación de estos árboles tiene como objetivo generar un modelo que prediga el valor de una variable de destino en función de una serie de variables de entradas.

Los árboles de decisión están formados por atributos y clases. Cada ramificación del árbol posee un conjunto de atributos asociadas a una etiqueta de clase específica. Para entrenarlo,

DESCRIPCIÓN DE LAS TECNOLOGÍAS

son necesarios tantos datos como atributos y la clase a la que ese vector de datos corresponde. Los datos se registran de la siguiente manera:

$$(x, Y) = (x_1, x_2, x_3, \dots, x_k, Y)$$

Siendo Y la variable objetivo que se quiere clasificar, y el vector x las variables de entrada.

Capítulo 3. DEFINICIÓN DEL TRABAJO

3.1 JUSTIFICACIÓN

Como se ha mencionado hasta ahora, este proyecto se realiza para detectar cuáles son las características más representativas de la voz y qué técnica (de las comparadas) es la que mejor realiza la automatización del reconocimiento del estado de ánimo que posee una persona en un momento determinado.

El tema del reconocimiento de la voz no es nuevo, como hemos podido ver en el apartado del [Estado del arte](#), pero existen únicamente algunas aplicaciones comerciales cerradas y muchas técnicas de inteligencia artificial aplicables. Es interesante poder conocer cuál es la solución que realiza un mejor reconocimiento de voz para los ámbitos en los que se podría comercializar el proyecto. A continuación, se plantean una serie de puntos en los que su aplicación podría ser útil, llegando a generar un producto que se pudiera comercializar.

3.1.1 CALL CENTERS

La automatización del reconocimiento de la voz puede ser útil para, por ejemplo, una empresa que disponga de un servicio telefónico de atención al cliente automatizado. En este tipo de servicios, al no tener de manera presencial al cliente y estar interactuando con un sistema informático, es muy complicado aprovechar la respuesta que da el cliente con sus emociones en el trato con la máquina para el proceso que se esté llevando a cabo. A continuación, se exponen una serie de casos en los que se podría aplicar:

DEFINICIÓN DEL TRABAJO

- Venta telefónica automática de productos: el sistema puede detectar el interés (o no) de un potencial o actual cliente por la llamada en curso, las áreas de interés que se le van ofreciendo y los productos correspondientes.
- Control del proceso de venta telefónica realizada por personas: el sistema puede controlar la evolución del discurso del cliente y del operador, para proponer mejoras o cambios.
- Soporte telefónico: en caso de averías, ante un sistema que por ejemplo tenga un árbol de decisión para ayudar al cliente con un problema o avería se puede detectar en los pasos que va dando el cliente si el proceso está siendo correcto desde el punto de vista del cliente o le está generando algún tipo de enfado, miedo o tristeza.

Estos tres puntos permiten:

- Un aumento de las ventas de los productos, gracias a un mejor proceso de automatización de la venta o del control de la venta realizada por personas.
- Una disminución de clientes perdidos por “un mal trato” telefónico.

Por ello, a través de la automatización del reconocimiento de emociones, una empresa puede ser capaz de generar una serie de datos con los que se pueden establecer estrategias para la toma de decisiones que beneficien a la empresa.

3.1.2 SEGURIDAD

En el ámbito de la seguridad este tipo de sistemas podrían complementar a los actuales sistemas de reconocimiento de voz para el acceso a determinados sitios, permitiendo de esta manera evitar accesos no autorizados por estar siendo coaccionados. Es un ámbito que podría aplicar tanto a las grandes entidades (empresariales, como bancos, u públicas, como acceso a edificios de órganos de gobierno o justicia), y también ampliarse al entorno de la seguridad privada “pequeña”, de los pequeños negocios y el hogar, que actualmente no hacen uso de

DEFINICIÓN DEL TRABAJO

las tecnologías de voz, pero que podrían verse muy beneficiadas (por ejemplo, lanzar una alarma ante una detección de miedo en el habla del propietario y agresividad en el de una persona ajena).

3.1.3 HOGAR Y AUTOMÓVIL

Dentro del hogar y de los vehículos privados, hoy en día están empezando a aparecer aplicaciones con las que interactuamos con la voz. Son aplicaciones orientadas a la ayuda de la persona, tanto en el hogar como mientras conduce. En ambos ámbitos, el reconocimiento de emociones de la voz permitiría aplicaciones avanzadas, como detección de una situación problemática en el hogar (tanto para la persona como de frustración ante el manejo del propio sistema), o ayuda en la localización de sitios dentro de un sistema de navegación (desde la sensación de no hay manera de encontrar el sitio al que quiero ir, hasta un sistema que vaya ayudando a encontrarlo).

Bajo estos casos (hogar y automóvil), el sistema de reconocimiento de voz previsiblemente no se percibiría como un valor añadido por parte del usuario por el que estuviera de acuerdo en pagar específicamente (salvo en los casos de seguridad o sanidad, de los que se habla en otros apartados), sino que serían las empresas propietarias de los productos en uso las que lo harían, para mejorar el servicio hacia sus clientes.

3.1.4 SANIDAD

En los hospitales, están comenzando a aparecer guías informativas para que cuando llegemos, que normalmente es agobiados o con cierto malestar, podamos encontrar fácilmente al lugar al que tenemos que acudir. Podría ser beneficioso para una buena atención a los usuarios que este sistema incorporará el reconocimiento de emociones,

DEFINICIÓN DEL TRABAJO

pudiendo el médico que les va a atender conocer previamente el estado de ánimo con el que vienen, y así poder ser atendidos de manera óptima.

Otra aportación que se podría hacer en este ámbito sería ayudar al colectivo autista o con síndrome de Asperger [23]. Estas personas tienen dificultades a la hora de reconocer emociones por ellas mismas, pero incorporando este sistema a su día a día podrían mejorar la forma de relacionarse con su entorno.

3.2 OBJETIVOS

Los objetivos del proyecto van a ser dos:

1. Detectar cuáles son las características de la voz relevantes para el reconocimiento de emociones.
2. Comparar el reconocimiento que es capaz de hacer un sistema open source existente basado en análisis vectorial, versus un sistema que aplica tecnologías de redes neuronales artificiales (mapas autoorganizados) y redes neuronales artificiales complementadas con árboles de decisión.

Evidentemente, existen muchas más herramientas y técnicas que se podrían utilizar en el proyecto, pero se dejarían para trabajos futuros dependiendo de las conclusiones a las que se lleguen.

3.3 METODOLOGÍA

Este proyecto se ha dividido en seis fases:

DEFINICIÓN DEL TRABAJO

Trabajos previos	Solución EmoVoice	Conclusiones
	Solución SOM	
ER	Solución SOM con árboles de decisión	

Para la realización del proyecto se ha utilizado una metodología de desarrollo software tradicional (especificación de requisitos, análisis y diseño, implementación, pruebas), que se ha seguido para cada una de las soluciones a completar, añadiendo una fase de trabajos previos y otra de conclusiones finales comunes. Se ha considerado que por homogeneidad que la fase de definición de requisitos también debía ser común.

Se pueden ver más detalles de la metodología y su planificación en el apartado [3.4 Planificación y Estimación económica](#). En los siguientes apartados se hace una descripción de la metodología utilizada en cada una de las fases más relevantes (trabajos previos y soluciones).

3.3.1 FASE TRABAJO PREVIO

Dentro de los trabajos previos a las soluciones, se plantea realizar una investigación de herramientas open source en el ámbito del proyecto, y de técnicas de inteligencia artificial aplicables (ver [1.2.2 Técnicas de clasificación de datos o reconocimiento de patrones](#)), así como un trabajo de definición y preparación de las muestras.

DEFINICIÓN DEL TRABAJO

En lo referente a la preparación de las muestras, se realiza una familiarización con la herramienta *openSMILE*. Como dicha herramienta requiere un archivo de configuración para su implementación, se ha utilizado el archivo que utiliza *EmoVoice* para que todas las técnicas de clasificación trabajen bajo las mismas condiciones y se puedan establecer conclusiones más sólidas. Dicho archivo de configuración se adjunta en el [Anexo A](#).

Después de conocer bien la herramienta, surge la necesidad de obtener una gran cantidad de audios que puedan servirnos para entrenar las tecnologías mencionadas en el [Capítulo 2](#). De aquí surge una búsqueda de una base de datos con múltiples audios que expresen emociones reales. La mejor base de datos encontrada es la que se emplea en este documento, esta trata de una base de datos llamada RAVDESS DATABASE [24]. Esta base de datos está formada por un total de 1.306 audios que expresan diferentes emociones. Estos audios fueron grabados por un grupo de 24 actores siendo la duración de cada audio de unos 3 segundos. De entre todas estas muestras se eligieron aquellas que pertenecían a las emociones *enfado*, *tristeza*, *asco*, *sorpres*a y *neutral*. Se optó por estas emociones principalmente porque como se le quiere dar un enfoque de reconocer emociones para saber cómo ha sido la atención al cliente y éstas son las más idóneas para ello.

Una vez obtenida la base de datos, primero se realizó una separación de las muestras de cada emoción de la base de datos para tener un conjunto para el entrenamiento dejando otro conjunto de datos exclusivamente para las pruebas para que fueran completamente nuevas para cada uno de los métodos de clasificación.

Tras dicha separación, se realiza la extracción de las características de cada uno de los audios. Para ello, se utilizó *openSMILE* junto con el archivo de configuración antes mencionado. El resultado fue un archivo CSV por cada muestra que contiene las características de unos 380 frames por audio. Con todos estos archivos CSV se produjo un archivo EXCEL por cada emoción. Estos EXCEL están formados por las características de

DEFINICIÓN DEL TRABAJO

6 frames de cada uno de los audios. Al llevarlo a un archivo EXCEL el manejo de los datos va a ser mucho más fácil para llevarlo a la entrada de las técnicas en las que pueda ser necesario.

3.3.2 FASES DE CADA SOLUCIÓN

Las fases de solución EmoVoice, SOM y SOM junto a árbol de decisión se realizan siguiendo una metodología tradicional software en la que:

1. Se analizan los requisitos considerando la técnica a aplicar.
2. Se prepara el software necesario.
3. Se ejecuta una fase de aprendizaje con las muestras correspondientes, y aplicando el proceso que se defina para cada una de las soluciones.
4. Se ejecuta una fase de reconocimiento de nuevas muestras, obteniendo para cada una de ellas la emoción que intenta transmitir.
5. Se intenta realizar una extracción de conclusiones sobre los resultados de las nuevas muestras, y dependiendo de las mismas, se realizan nuevas iteraciones de ejecución.

3.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

3.4.1 PLANIFICACIÓN

La planificación final del proyecto queda como se muestra en la *Ilustración 1*:

DEFINICIÓN DEL TRABAJO

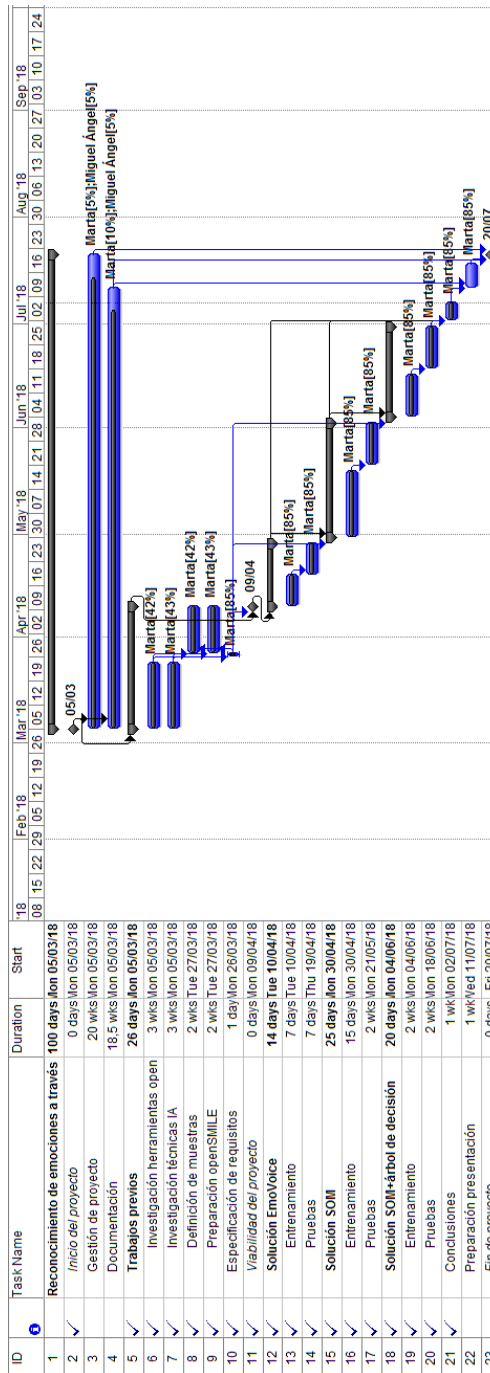


Ilustración 1. Planificación proyecto

DEFINICIÓN DEL TRABAJO

Se ha tenido en cuenta para la elaboración del proyecto un calendario con los únicos festivos en la semana de los sábados y los domingos, y no otras festividades.

Para todas las tareas, se han asignado los recursos humanos necesarios para poder realizarlas en un porcentaje que en ningún caso exceda el 100% para las tareas que se puedan ejecutar en paralelo.

En la fecha de entrega de esta memoria, únicamente quedará por completar la presentación del proyecto y el cierre del mismo.

3.4.2 ESTIMACIÓN ECONÓMICA

Desde un punto de vista económico, el coste de este proyecto es el siguiente:

Actividades	Coste
Gestión de proyecto	1.320,00 €
Documentación	1.702,00 €
Trabajos previos	2.210,00 €
Especificación de requisitos	88,40 €
Solución EmoVoice	1.237,60 €
Solución SOM	2.210,00 €
Solución SOM + árbol de decisión	1.768,00 €
Conclusiones	442,00 €
Preparación presentación	442,00 €
Total	11.420,00 €

En este coste se ha incluido el coste del personal del proyecto. No se han considerado los costes de las instalaciones ni la compra de ningún elemento hardware adicional, puesto que se contaba con el material suficiente.

Por último, se ha realizado un pequeño análisis de riesgos del proyecto:

DEFINICIÓN DEL TRABAJO

ID	Riesgos	Plan de contingencia	Acciones de contingencia	Sobrecoste
1	Los resultados de algún método no sean los esperados.	Mitigar/ Explotar	1. Elaborar un estudio previo de la tecnología 2. Realizar pruebas alternativas variando los parámetros inicialmente definidos.	520 €
2	Algún módulo open-source de los utilizados deje de estar disponible como tal.	Evitar	Utilizar open-sw reconocido en Internet y usado por muchos usuarios.	0 €
3	Algún recurso no se encuentre disponible temporalmente por exámenes u otro tipo de ausencias.	Aceptar	Replanificar la fecha de entrega.	0 €
Total: 520 €				

En caso de que el proyecto se vendiera a un tercero, sería necesario estimar el HW en el que se va a entregar, así como las licencias y adaptaciones de interfaz. Asimismo, debería incluirse el coste estimado de las contingencias, valorando económicamente también el riesgo de retrasos del proyecto.

Capítulo 4. REQUISITOS DEL PROYECTO

4.1 REQUISITOS GENERALES

GEN.1 Objetivo de cada solución a implementar

La solución que se implemente deberá ser capaz de dada una muestra de audio, indicar la emoción que más probablemente se está intentando transmitir.

GEN.1.1 Formatos de las muestras de audio

Las muestras de audio estarán codificadas en PCM WAVE (.WAV) monomodo.

GEN.1.2 Duración de las muestras de audio

La duración de las muestras de audio tiene que ser la misma en el entrenamiento y las pruebas.

GEN.1.3 Emociones consideradas

Las muestras podrán tener una de estas emociones: *enfado, asco, tristeza, sorpresa y neutral.*

GEN.2 Aprendizaje de cada solución a implementar

La solución que se implemente deberá ser capaz de aprender de un conjunto de muestras de audio.

GEN.2.1 Tamaño del conjunto de muestras de audio para aprendizaje

REQUISITOS DEL PROYECTO

El tamaño del conjunto de muestras de audio para aprendizaje será siempre de 900 muestras para todas las soluciones que se implementen.

GEN.3 Reconocimiento de emociones

Una vez una solución realice el aprendizaje de emociones, se le podrá presentar una muestra nueva, para la que indicará la emoción más probable.

GEN.4 Soluciones a implementar

Se implementarán varias soluciones que realicen un reconocimiento de emociones de la voz.

GEN.4.1 Tipos de soluciones

Se implementará una solución que utilice una red neuronal, una solución de análisis vectorial, y una solución que utilice una red neuronal con un árbol de decisión.

GEN.4.2 Autoaprendizaje

En los casos de redes neuronales, como parte del aprendizaje no se indicará la emoción representada por la muestra de aprendizaje.

4.2 REQUISITOS DE INTERFAZ

UI.1 Interfaz de usuario

Las soluciones se podrán invocar por línea de comandos o desde una aplicación, indicando el conjunto de muestra de aprendizaje, por un lado, o la muestra a reconocer por otro.

UI.2 Salida como resultado del reconocimiento de una emoción

La solución informará al usuario de la emoción más probable para cada muestra a reconocer.

4.3 REQUISITOS DE HARDWARE

HW.1 Requisitos HW para las soluciones

Todas las soluciones que se implementen deberán poder ejecutarse de forma independiente en un ordenador portátil de consumo.

Capítulo 5. SOLUCIÓN EMOVOICE

Como se explica en el [apartado 2.2](#), EmoVoice es una herramienta que permite la creación de un reconocedor de emociones propio a partir de las propiedades acústicas del habla. Esta solución integra principalmente los siguientes módulos:

1. La creación de una base de datos con muestras propias,
2. la solución *openSMILE* para la extracción de características,
3. y la construcción y evaluación de un clasificador basado en una máquina de soporte vectorial.

5.1 ANÁLISIS DEL SISTEMA

La arquitectura de este sistema está formada por una serie de subsistemas, los cuales son principalmente el software *openSMILE* ([2.1](#)), el cual requiere un archivo de configuración, y una máquina de soporte vectorial (SVM) que clasifica las muestras. Dicha arquitectura se muestra en la *Ilustración 2*.

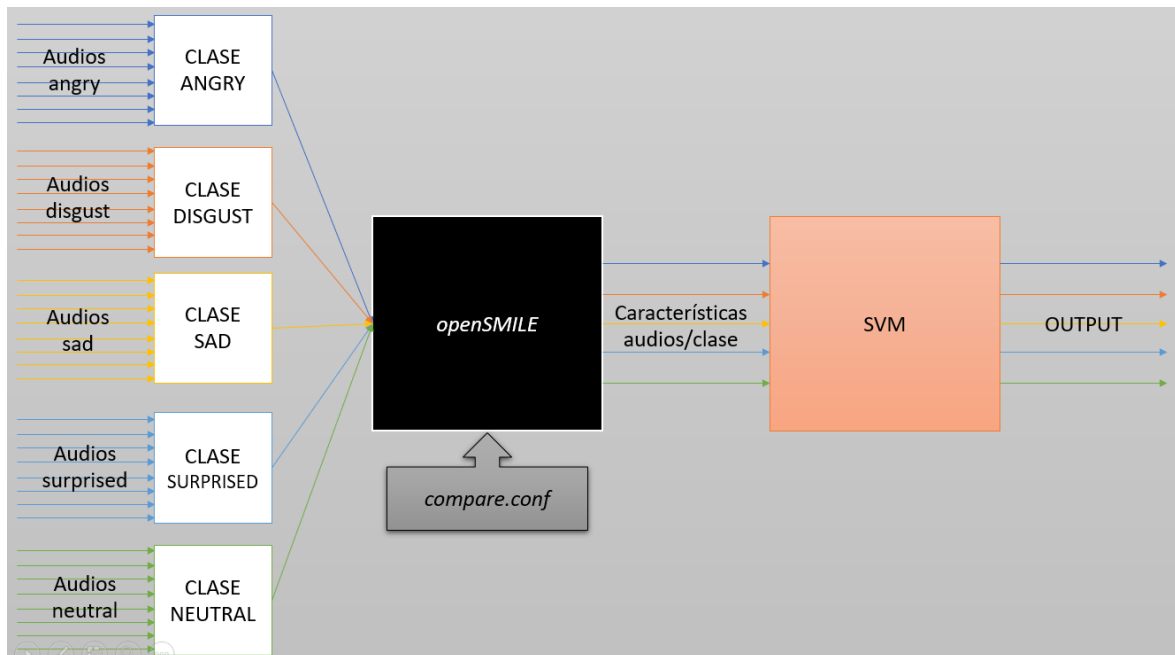


Ilustración 5. Arquitectura EmoVoice

En el siguiente apartado, se explica detalladamente el procedimiento seguido para la configuración de cada uno de los subsistemas.

5.2 IMPLEMENTACIÓN

La implementación de la solución EmoVoice consistió principalmente en el entrenamiento de la máquina de soporte vectorial con una serie de muestras de la base de datos RAVDESS. Los siguientes apartados muestran el proceso seguido para dicha configuración.

5.2.1 PREPARACIÓN DE LOS DATOS

La preparación de los datos para el entrenamiento se realiza dividiéndose las muestras en diferentes clases, correspondiendo cada una de ellas a una emoción. Es necesario disponer

SOLUCIÓN EMOVOICE

de una base de datos con muchos audios de cada emoción para que el entrenamiento sea efectivo. El entrenamiento realizado con la base de datos utilizada incluye las emociones *enfado, asco, tristeza, sorpresa y neutral*.

El comando `do_vad` de EmoVoice prepara los datos dividiéndolos en las clases señaladas en el segundo argumento "angry;disgust;neutral;sad;surprised", las cuales van a tener que contener una serie de muestras con la misma frecuencia de muestreo, que en este caso vale 48.000 Hz.

```
> do_vad 48000 "angry;disgust;neutral;sad;surprised"
```

5.2.2 EXTRACCIÓN DE CARACTERÍSTICAS

Una vez se han indicado las diferentes clases que va a tener nuestro sistema, es necesario extraer las propiedades acústicas del conjunto de muestras. Para ello, se ejecuta el siguiente comando:

```
> do_samples compare
```

El argumento "compare" indica las características que se quieren extraer en dicho audio. *EmoVoice* también permite utilizar otros conjuntos de características (*emovoice* y *gemaps*).

Al señalarle "compare" como el conjunto de características con el que se quieren recoger las propiedades de los audios, se está indicando la utilización de la herramienta *openSMILE*. Como se explica en el [apartado 2.1](#), *openSMILE* necesita, a parte de los audios, un archivo de configuración para la extracción de características. El archivo utilizado en *EmoVoice* para la configuración de *openSMILE*, el cual se adjunta en el [Anexo A](#), realiza un tratamiento de la señal de audio a través de un conjunto de componentes definidos por *openSMILE*. Entre las características que se extraen se encuentran los coeficientes MFCC, el tono, la calidad de

la voz y los descriptores espectrales. A estas características se las ha aplicado unas funciones estadísticas (rango, máximo, mínimo, percentiles, etc.).

El resultado de ejecutar este comando es una lista de muestra para cada clase almacenada con el conjunto de características indicado, que como se ha mencionado en este caso es "compare".

5.2.3 FASE DE ENTRENAMIENTO

En esta fase se procede a entrenar el modelo de clasificación basado en SVM (Máquina de Soporte Vectorial). Para ello se ejecuta el siguiente comando:

```
> do_train linsvm compare
```

El primer argumento indica el nombre del clasificador que en este caso es "linsvm" y el segundo argumento es el nombre del conjunto de características utilizado "compare".

De esta manera se mapean los vectores de características del conjunto de clases discretas establecido en fases anteriores.

Para poder evaluar la calidad del modelo de clasificación creado con las muestras utilizadas se utiliza el comando siguiente:

```
> do_eval linsvm compare
```

Con este comando se obtiene una matriz de confusión que muestra la precisión de cada clase y la tasa de reconocimiento general de nuestro modelo de clasificación:

```
#classes:      5  
#features:     6373  
#total:        704  
#classified:   704
```

SOLUCIÓN EMOVOICE

```

#unclassified: 0

angry:      115    16     2     6    12    ->    76.16
disgust:    17     78    28    24    15    ->    48.15
neutral:    1      5     84    20     6    ->    72.41
sad:        5     16    35    67     6    ->    51.94
surprised:  6      6     7     21   106    ->    72.60
                                     =>    64.25 | 63.92
  
```

Como se puede apreciar, la clase con una clasificación más fiable es la de *angry*, con un porcentaje de 76.16 % y la emoción con peores resultados es la de *disgust* con un 48.15 % que aun así tampoco es una cifra demasiado baja. En su conjunto, la clasificación no ha tenido unos resultados óptimos, pero son ciertamente altos con un 64.25 % de probabilidad de acierto en general.

Esta evaluación la realiza el propio sistema *a priori*, en la fase de pruebas se mostrarán los resultados obtenidos una vez clasificadas una serie de muestras completamente nuevas para el sistema.

5.2.4 FASE DE RECONOCIMIENTO

Tras realizar el entrenamiento del modelo, si la calidad de este es buena se podrá proceder al reconocimiento de emociones. Para ello, como se ha mencionado anteriormente, se introducirá una muestra nueva a la cual se le aplicará la detección de actividad de voz para encontrar los segmentos de voz y se le extraerán las características. Una vez extraídas las características se observará a qué clase corresponde de las establecidas en las fases anteriores según la similitud de las propiedades del nuevo audio a los usados en el entrenamiento. De esta manera, se concluirá la emoción que expresa la muestra introducida.

En la *Ilustración 3* se puede observar cómo es el procesamiento explicado de la señal de audio cuando es tratado por *EmoVoice*.

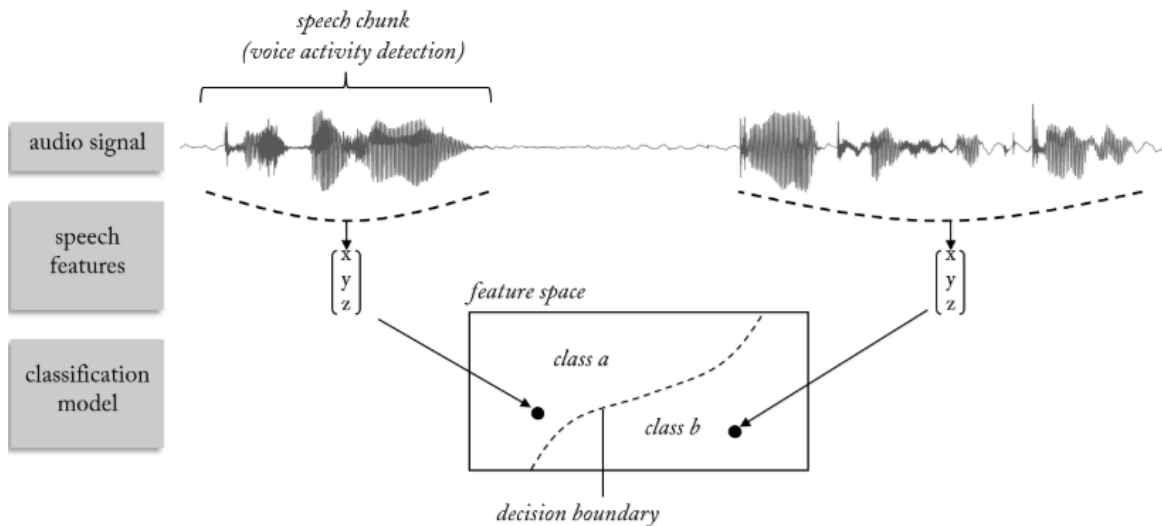


Ilustración 6. Procesamiento de la señal de audio a través de EmoVoice

5.3 ANÁLISIS DE PRUEBAS

Tras realizar el entrenamiento explicado en el apartado anterior, se realiza una fase de pruebas *testeando* una colección de muestras nuevas. EmoVoice, tras extraer las características y situar los nuevos ejemplos en una de las cinco clases, muestra el resultado a través de la interfaz mostrada en la *Ilustración 4*.

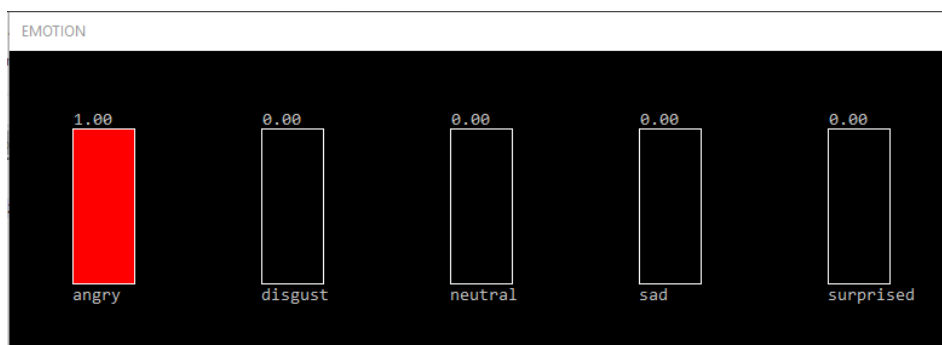


Ilustración 7. Interfaz EmoVoice

SOLUCIÓN EMOVOICE

Los resultados obtenidos se muestran en la *Tabla 1*. Se puede afirmar que las características de la emoción *enfado* distan de las demás ya que, según la matriz, no existe confusión con el resto de las emociones. Otro dato a destacar es que, además de la clase *enfado*, la emoción *neutral* y *surprised* son las que tienen mayor porcentaje de acierto. La clase con mayor error es la de *asco* ya que posee un porcentaje de acierto del 63.1579 %, lo que conlleva a un error del 36.8421 % superando al resto de clases en este aspecto.

El resultado del conjunto es que la máquina de soporte vectorial es una técnica bastante fiable ya que los porcentajes obtenidos son altos, con un margen de error relativamente pequeño.

		Muestra clasificada como ->				
		A	D	N	SA	SU
Emoción de la muestra ->	A	100 %	0 %	0 %	0 %	0 %
	D	10.5263 %	63.1579 %	10.5263 %	15.7895 %	0 %
	N	0 %	5.2632 %	89.4737 %	5.2632 %	0 %
	SA	5.2632 %	10.5263 %	5.2632 %	73.6842 %	5.2632 %
	SU	5.2632 %	5.2632 %	0 %	0 %	89.4737 %

Tabla 1. Matriz de confusión pruebas EmoVoice.

(A = enfado, D = asco, N = neutral, SA = tristeza, SU = sorpresa).

Capítulo 6. SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

El mapa autoorganizado, como se ha mencionado en el [apartado 2.3](#), es un modelo de red que se caracteriza por poseer un aprendizaje no supervisado competitivo.

6.1 ANÁLISIS DEL SISTEMA

En este proyecto, se han construido cinco mapas autoorganizados, uno por cada emoción, tomando como entrada las características de los audios de muestra. Tras el trabajo previo realizado (ver [Fase trabajo previo 3.3.1](#)), se prepararon los datos de cada emoción en archivos EXCEL que nos sirven como entrada al mapa autoorganizado. Como dichos archivos están formados por demasiados datos, se realiza una división para poder observar y manejar mejor los resultados del SOM. Esta separación en bloques de las características se muestra en el [Anexo B](#).

Cada uno de los modelos SOM utilizados en este proyecto consta de una única capa de neuronas. La información sigue el siguiente flujo: los datos de entrada son transmitidos a la capa de neuronas (formada por 9 neuronas), que a su vez transmiten la información a la salida compuesta por 9 outputs formando así el mapa. De esta manera se reducen los datos de un espacio de mayor dimensión a uno de menor dimensión. En la *Ilustración 6* se muestra el flujo descrito que sigue la información de cada SOM, siendo N el número de entradas que varía según el conjunto de características.

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

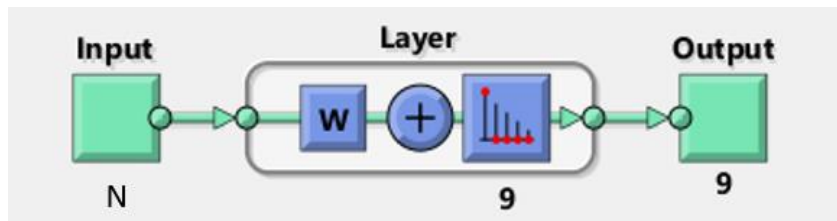


Ilustración 8. Arquitectura SOM

Con la realización de estos SOMs se trata de observar las diferencias de dichos mapas para identificar las características de la voz más útiles para identificar una determinada emoción. Tras realizar la fase de pruebas (fase de mapeo) se observará la fiabilidad que posee tanto esta técnica como los datos introducidos para el reconocimiento de emociones.

6.2 IMPLEMENTACIÓN SOMs

6.2.1 CODIFICACIÓN

El SOM se implementa gracias a la herramienta *Matlab*. A continuación, se muestra el código fuente del mapa autoorganizado para poder mapear las características de los audios:

```
clear all
close all
%%
% Carga de datos:
%datos = xlsread('angry','Hojal','IJP2:IKD901'); %ANGRY
%datos = xlsread('disgust','Hojal','IJP2:IKD900'); %DISGUST
%datos = xlsread('sad','Hojal','IJP2:IKD907'); %SAD
%datos = xlsread('surprised','Hojal','IJP2:IKD901'); %SURPRISED
datos = xlsread('calm_neutral','Hojal','IJP2:IKD901'); %CALM_NEUTRAL

input=datos';

[natrib nn]=size(input);
%%
```

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

```
% Se construirá un mapa de dimension1xdimension2

dimension1 = 3; %anchura mapa - dimension X
dimension2 = 3; %Altura mapa - dimension Y

load('sm'); % aquí simplemente hay una estructura que facilita luego
gráficos
sm.topol.msize=[dimension2 dimension1]; % En esa estructura se actualiza
dimensiones

iteraciones = 2000;
radio_v=2;
top='gridtop';
net = selforgmap([dimension1 dimension2], iteraciones,radio_v, top);

net.trainParam.lr = 0.6; %Factor de aprendizaje

% Entrenamos la red
[net,tr] = train(net,input);

pesos= net.IW{1};

figure(10)
for i=1:dimension1*dimension2
    plot(1:natrib,pesos(i,:))
    hold on
end
title('Patterns of data')
xlabel('Attributes')
ylabel('Values of weights')
legend
hold off

%Se reordenan las posiciones de pesos para usar unas funciones auxiliares
%de graficado

pesos1=pesos;

m=0;
for j=1:dimension2
    for i=1:dimension1
        ff = pesos(i+dimension1*m, :, :);
        pesos1(i*dimension2 - m, :, :) = ff;
```

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

```

end
m=m+1;
end

sm.codebook= pesos1; % Actualización de pesos reordenafos

% Empiezan funciones auxiliares de graficado
figure(6)
T = som_normalize(sm.codebook, 'range'); % Para normalizar
%T=pesos1; %Esto no normaliza
som_barplane(sm, T, '', 'unitwise');
title('SOM\_patterns')

figure(7)
som_pieplane(sm, T);
title('SOM\_patterns')

figure(8)
som_plotplane(sm, T, 'b');
title('som\_patterns')

%%
y=net(input);

```

6.2.2 FASE DE ENTRENAMIENTO

A continuación, se van a analizar los mapas que resultan del entrenamiento del conjunto de características del *bloque1_1* a modo de ejemplo del proceso seguido. En la *Ilustración 7* se pueden observar las distintas gráficas de los SOMs de cada emoción (para mejor visualización ver [Anexo C](#)). En términos generales, la *gráfica SOM1* muestra las neuronas del SOM, éstas se enumeran de izquierda a derecha empezando por la neurona de abajo a la izquierda – siendo la neurona 1 – hasta llegar a la de arriba a la derecha – siendo la neurona 9. Cada una de las gráficas de la *gráfica SOM2* corresponden a cada neurona y representan los patrones de los pesos normalizados de los datos de entrada. Y, por último, la *gráfica SOM3* representa los pesos de los datos introducidos al SOM de cada una de las neuronas.

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

De estas gráficas se analizará la correspondiente a la neurona activada, es decir, en la que han caído el mayor número de muestras. Dicha neurona nos va a mostrar los patrones que más definen a cada emoción.



Ilustración 9. SOMs características bloque1_1.

De arriba abajo: gráfica SOM1, gráfica SOM2 y gráfica SOM3.

Como se ha mencionado, en las siguientes ilustraciones se muestran las gráficas de las neuronas activadas de cada SOM. Un aspecto importante que se va a analizar es la existencia de cierta diferencia entre los patrones de los diferentes SOMs, porque de no existir quiere decir que no hay forma de realizar una clasificación de las emociones a partir de las características utilizadas.

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

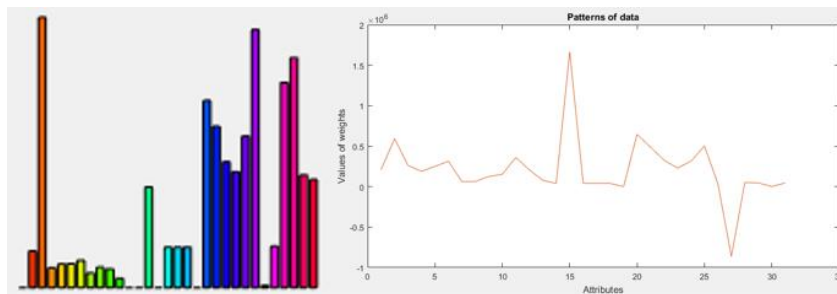


Ilustración 10. SOM Angry Neurona 9.

De izquierda a derecha: gráfica 1 y gráfica 2.

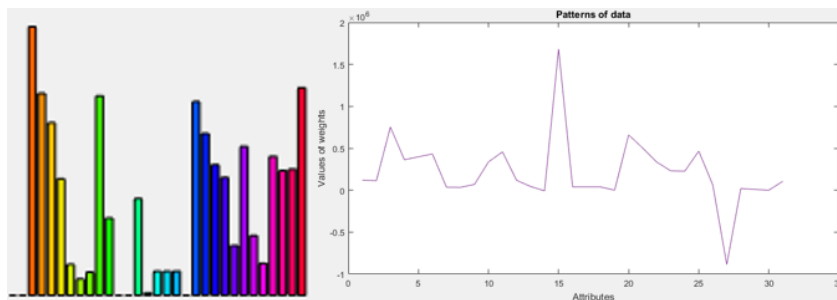


Ilustración 11. SOM Disgust Neurona 4.

De izquierda a derecha: gráfica 1 y gráfica 2.

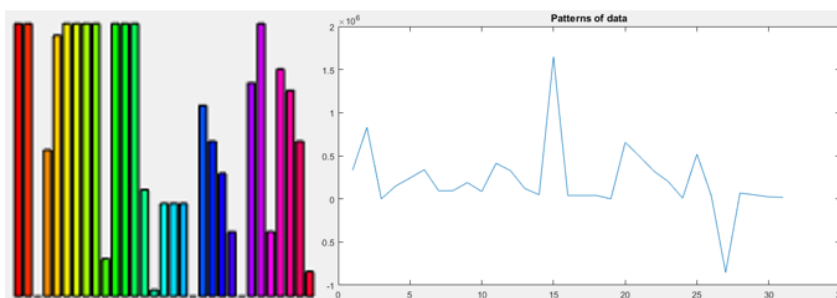


Ilustración 12. SOM Sad Neurona 1.

De izquierda a derecha: gráfica 1 y gráfica 2.

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

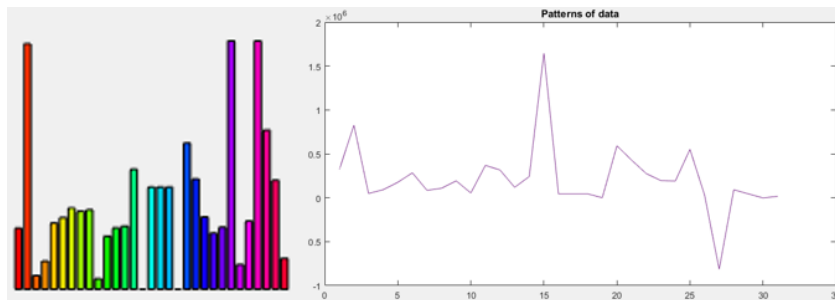


Ilustración 13. SOM Surprised Neurona 4.

De izquierda a derecha: gráfica 1 y gráfica 2.

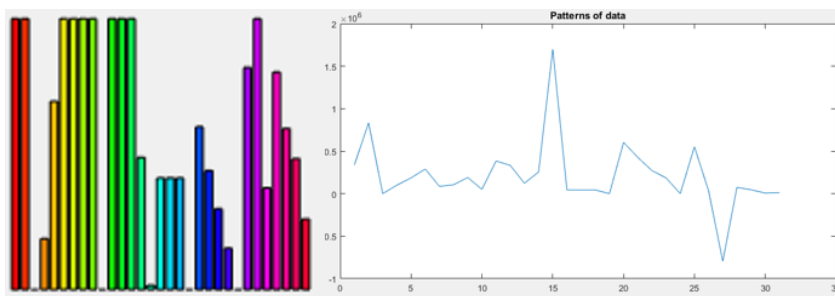


Ilustración 14. SOM Neutral Neurona 1.

De izquierda a derecha: gráfica 1 y gráfica 2.

En las *Ilustraciones 8, 9, 10, 11 y 12* se pueden observar las gráficas correspondientes a la neurona activada de cada SOM – la neurona 9 en el SOM *enfado*, la neurona 4 en el de *asco*, la neurona 1 en el de *tristeza*, la neurona 4 en el de *sorpresa* y la neurona 1 en el de *neutral*, respectivamente.

Entre las *gráficas 1* de cada SOM, las cuales, como se ha mencionado, representan los pesos normalizados de los datos de entrada, muestran cierta redundancia las características 10, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 27 y 30, ya que son muy similares entre sí. En el resto de los patrones de las características sí que hay diferencia de un SOM a otro. Entre las *gráficas 2* de los SOMs, se muestra una representación muy similar pero que en determinadas

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

características hay más diferencia. Esta gráfica nos indica en todos los SOMs que el centro de gravedad de los pesos se encuentra en torno a la característica 15.

Este proceso de análisis se ha seguido tomando como entrada cada conjunto de características. Para comprobar la fiabilidad de estos SOMs para diferenciar emociones, se han realizado una serie de pruebas, las cuales se describen en el apartado de [Análisis de pruebas 6.3](#).

6.2.3 FASE DE RECONOCIMIENTO

La fase de clasificación se realiza utilizando los mapas obtenidos de la fase de entrenamiento. Para realizar las pruebas se utiliza la función `sim(net, input)` de Matlab, que sirve para realizar una simulación sin modificar la red. Esta función tiene como argumentos:

- **Primer argumento `net`:** la estructura net correspondiente al SOM en el que se quiera evaluar la nueva muestra.
- **Segundo argumento `input`:** las características de la nueva muestra que, al igual que en el entrenamiento de los SOMs, están en un formato EXCEL.

La salida de la función `sim` es un vector. Este vector tiene longitud 9 y el contenido de cada elemento del vector corresponde a una neurona. Sólo uno de los 9 elementos va a valer 1, el cual corresponde a la neurona activada por la nueva muestra. Si la neurona en la que ha caído la muestra en el proceso de pruebas coincide con la neurona con mayor peso (o entre las que tienen mayor peso) significará que esa muestra tiene muchas probabilidades de pertenecer a ese SOM, significando a su vez que sus patrones son muy similares. De esta manera, se identificará la emoción a la que pertenece la nueva muestra.

Las pruebas mencionadas se han realizado al igual que los SOMs con la herramienta Matlab. El código de pruebas es el siguiente:

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

```
clear all
close all

datos = xlsread('pruebas_angry\test03-01-02-02-01-01-02','test03-01-02-02-01-01-02','B94:AF157'); %PRUEBA

input=datos';

load('net_angry_1_1');
P_angry = sim(net, input);
for i=1:9
    n_angry(i)=0;
    for j=1:length(P_angry)
        n_angry(i)=n_angry(i)+P_angry(i,j);
    end
end
n_angry=n_angry';

load('net_disgust_1_1');
P_disgust = sim(net, input);
for i=1:9
    n_disgust(i)=0;
    for j=1:length(P_disgust)
        n_disgust(i)=n_disgust(i)+P_disgust(i,j);
    end
end
n_disgust=n_disgust';

load('net_sad_1_1');
P_sad = sim(net, input);
for i=1:9
    n_sad(i)=0;
    for j=1:length(P_sad)
        n_sad(i)=n_sad(i)+P_sad(i,j);
    end
end
n_sad=n_sad';

load('net_surprised_1_1');
P_surprised = sim(net, input);
for i=1:9
    n_surprised(i)=0;
```

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

```
for j=1:length(P_surprised)
    n_surprised(i)=n_surprised(i)+P_surprised(i,j);
end
end
n_surprised=n_surprised';

load('net_neutral_1_1');
P_calmNeutral = sim(net, input);
for i=1:9
    n_calmNeutral(i)=0;
    for j=1:length(P_calmNeutral)
        n_calmNeutral(i)=n_calmNeutral(i)+P_calmNeutral(i,j);
    end
end
n_calmNeutral=n_calmNeutral';
```

Como cada muestra se pasa por cada uno de los SOMs, el resultado de estas pruebas son 5 vectores por muestra: `n_angry`, `n_disgust`, `n_sad`, `n_surprised` y `n_calmNeutral`. Cada uno de los elementos de estos vectores son la suma de los elementos de cada neurona de la salida de la función `sim`, esta suma en realidad es un recuento de las veces que ha caído la muestra en una determinada neurona.

Con este recuento se ha realizado una matriz para cada conjunto de características de cada bloque, pudiendo saber la fiabilidad que tiene esta técnica para reconocer emociones.

Los resultados se pueden consultar en el siguiente apartado.

6.3 ANÁLISIS DE PRUEBAS

Las pruebas realizadas con la solución de los mapas autoorganizados consisten en introducir una colección de muestras completamente nueva (que no hayan formado parte del entrenamiento de los mapas) a cada uno de los SOMs para observar la neurona que se activa en este caso. Si la neurona activada coincide con la neurona que permanece activa en cada

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

uno de los SOMs – la 9 en el SOM *enfado*, la 4 en el SOM *asco*, la 1 en el SOM *tristeza*, la 4 en el SOM *sorpres*a y la 1 en el SOM *neutral* – quiere decir que dicha muestra pertenece a esa emoción.

Las siguientes tablas muestran el porcentaje de veces que una muestra de audio que expresa una determinada emoción que se quiere clasificar cae en la neurona activada de cada uno de los SOMs para el primero de los bloques de categorías/propiedades. Como se indica en la tabla, las filas corresponden a la emoción del audio testeado y las columnas a la emoción del SOM por el que se ha pasado dicha muestra.

Bloque1_1		SOM				
		A	D	SA	SU	N
AUDIOS	A	5.26 %	100%	0%	0%	100%
	D	52.63%	68.42%	5.26 %	0%	6
	SA	47.36%	15.78%	0%	5.26 %	5.26 %
	SU	26.31%	84.21%	0%	0%	26.31%
	N	100%	5.26 %	10.52%	0%	10.52%

Tabla 2. Matriz Bloque 1_1.

A = *enfado*, D = *asco*, SA = *tristeza*, SU = *sorprendido*, N = *neutral*

El resultado de la *Tabla 2*, correspondiente a los SOMs de las subcaracterísticas del bloque1_1, como se puede observar, demuestra muy poca fiabilidad ya que la probabilidad de acierto total es de un 16.84% junto con una probabilidad de fallo considerable. Existe mucha confusión, en el caso de las muestras de la emoción *enfado* las clasifica como *asco* y *sorpres*a en un muy elevado porcentaje. Las muestras *asco* son las mejor clasificadas con un 68.42% de aciertos, aunque con cierto margen de error también. Los ejemplos de la emoción *tristeza* son confundidos en un 47.36% con la emoción *enfado* y las muestras de *sorpres*a son clasificadas con un 84.21% como *asco*. Y, por último, siguiendo el error de las otras

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

muestras testeadas, las muestras de *neutral* son confundidas en un 100% de los casos con la emoción *enfado*.

Bloque1_2		SOM				
		A	D	SA	SU	N
AUDIOS	A	100 %	100 %	100 %	100 %	100 %
	D	100 %	100 %	100 %	100 %	100 %
	SA	100 %	100 %	100 %	100 %	100 %
	SU	100 %	100 %	100 %	100 %	100 %
	N	100 %	100 %	100 %	100 %	100 %

Tabla 3. Matriz Bloque1_2

A = enfado, D = asco, SA = tristeza, SU = sorprendido, N = neutral

La matriz obtenida tras realizar las pruebas con los SOMs correspondientes a las subcaracterísticas del bloque1_2 (ver *Tabla 3*), muestran una confusión total en la clasificación, siendo clasificada la muestra nueva con un 100% de probabilidades en todas las emociones, lo cual indica que estas características no aportan nada a la clasificación de emociones.

Bloque1_3		SOM				
		A	D	SA	SU	N
AUDIOS	A	0 %	0 %	5.2631 %	0 %	10.5263 %
	D	15.7894 %	0 %	15.7894 %	15.7894 %	26.3157 %
	SA	0 %	0 %	0 %	0 %	5.2631 %
	SU	0 %	0 %	0 %	0 %	5.2631 %
	N	0 %	0 %	0 %	0 %	15.7894 %

Tabla 4. Matriz Bloque1_3

A = enfado, D = asco, SA = tristeza, SU = sorprendido, N = neutral

En este caso, la *Tabla 4* nos muestra los resultados obtenidos tras realizar las pruebas con los SOMs de las subcaracterísticas del bloque1_3. Como se observa, la clasificación de la muestra nueva en cada SOM resulta con un 0 % de aciertos, con la excepción de la emoción *neutral* que posee un 15.7894 % de aciertos. En concreto, la muestra *enfado* se confunde con

SOLUCIÓN MAPA AUTOORGANIZADO (SOM)

la emoción *tristeza*, la muestra *asco* con mayor probabilidad se confunde con la emoción *neutral* y las muestras *tristeza* y *sorpresa* se confunden con la emoción *neutral*. Por lo tanto, el grado de confusión de esta clasificación es altísimo y no sirve para el reconocimiento de emociones.

Se puede observar que cada fila no suma el 100% ya que hay veces que algunas muestras caen en la neurona mayoritaria de más de un SOM.

Este proceso se ha realizado de la misma manera con el resto de los conjuntos de características. Los resultados siguen el mismo hilo que el ejemplo mostrado: muy baja fiabilidad con mucho error. Por lo tanto, dados estos resultados podemos afirmar que utilizar únicamente los mapas autoorganizados para reconocimiento de emociones, con una selección aleatoria de subconjuntos de características (debido a la gran cantidad de las mismas, ver [Anexo B](#)), no es un buen método. Más adelante se verá si complementando esta técnica con otra de aprendizaje supervisado mejoran los resultados o no, mediante entre otras cosas, una mejor selección de las características de entrada.

En el siguiente capítulo se explica el procedimiento seguido para el refinamiento de la fiabilidad de esta técnica realizando una serie de árboles de decisión que precisamente nos ayudan a extraer cuáles son las mejores características para el reconocimiento de emociones.

Capítulo 7. SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

En este capítulo se va a explicar más detalladamente el trabajo realizado con el árbol de decisión. Este árbol se ha empleado para refinar la fiabilidad de los resultados obtenidos con el SOM. Se van a utilizar los pesos extraídos de los SOMs de las cinco emociones para realizar una serie de árboles que nos permitan saber qué características tienen más importancia para reconocer emociones.

Para realizar dichos árboles de decisión se va a utilizar el software WEKA introduciéndole una serie de archivos ARFF de cada conjunto de características, que contienen los pesos de cada neurona de las distintas emociones. De esta manera se reducen los problemas de mínimos relativos cuando sólo se utiliza un SOM.

7.1 ANÁLISIS DEL SISTEMA

Este sistema es una continuación del mapa autoorganizado explicado en el [Capítulo 6](#). El sistema completo para llegar a la obtención de los árboles de decisión se muestra en la *Ilustración 13*. Este está formado por el software *openSMILE*, el SOM (creado con *Matlab*) y la herramienta WEKA. Como se ha ido explicando a lo largo del documento, primero se introducen los audios en *openSMILE* obteniendo así las características de cada audio. A continuación, dichas características se introducen en bloques en el mapa autoorganizado generando un conjunto de pesos de cada característica. Dichos pesos se introducen en el software WEKA en formato ARFF obteniendo como salida los árboles de decisión.

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

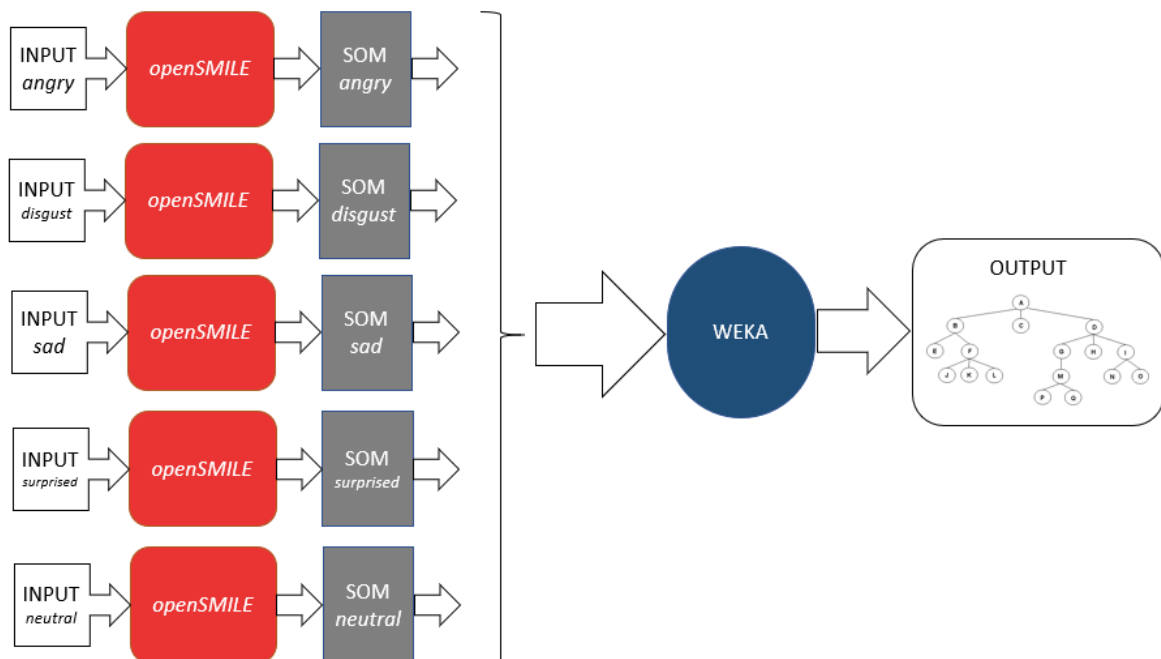


Ilustración 15. Arquitectura sistema completo

A través de los pesos extraídos por el SOM de cada una de las características, se pretende mejorar la fiabilidad, identificando las propiedades de la señal de la voz más importantes para el reconocimiento de emociones.

Se ha realizado un árbol de cada bloque de características, tal y como se hizo con la solución de los SOMs, de esta manera se identificará el bloque con mayor fiabilidad, y dicho árbol será el que tenga menor porcentaje de error.

Una vez terminado ese proceso (fase refinamiento SOMs), se comenzó otro en el que se redujeron los datos (fase reducción de datos). Consistió principalmente en realizar un único árbol de decisión con los pesos de unos nuevos SOMs. Estos nuevos mapas autoorganizados han sido entrenados con las características de los árboles de decisión del proceso anterior

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

que proporcionan mayor fiabilidad a la clasificación de emociones, excluyendo también las propiedades que no aparecen en dichos árboles.

7.2 IMPLEMENTACIÓN

Como se ha mencionado, para realizar los árboles se ha utilizado el software WEKA, introduciéndole una serie de archivos con formato ARFF. Estos archivos ARFF incluyen una serie de componentes:

- Los **atributos (nodos)**: corresponden a las características que se están empleando.
- Las **clases (hojas)**: son las emociones que se pretenden reconocer. En este proyecto, como se ha mencionado anteriormente son *enfado*, *asco*, *tristeza*, *sorprendido* y *neutral*.
- Los **datos (ramas)**: en esta parte del archivo se incluyen todos los pesos de cada uno de los atributos separados por comas y al final de cada fila se introduce una letra correspondiente a la emoción (A - *enfado*, D - *asco*, SA - *tristeza*, SU - *sorprendido* y N - *neutral*).

En el [Anexo D](#) se muestra un ejemplo de un archivo de este tipo.

El algoritmo utilizado para realizar los árboles de decisión se trata del J48, el cuál es una implementación open source en lenguaje de programación Java del algoritmo C4.5 [25]. Este algoritmo genera una estructura de reglas o árbol a partir del conjunto de datos de entrenamiento. Este proceso lo realiza en sucesivos ciclos. En cada uno de ellos, se incrementa la dimensión de la ventana de proceso relativa al conjunto total. Cada ciclo del proceso toma como punto de partida la salida del ciclo anterior. [26]

7.3 ANÁLISIS DE PRUEBAS

La metodología seguida varía ligeramente en esta solución. El proceso seguido con esta solución en la que se complementan dos técnicas se ha dividido en dos fases: una que realiza un refinamiento de la solución SOM, y otra que concluye las características más útiles para el reconocimiento de emociones. En ambas fases se realiza una subfase de entrenamiento y una subfase de reconocimiento de patrones nuevos, al igual que en las soluciones anteriores.

7.3.1 FASE REFINAMIENTO SOMs

Una vez realizados los documentos ARFF de cada uno de los bloques de características, se introducen en el software WEKA, extrayendo árboles como el que se muestra en la *Ilustración 14*.

Al realizar estos árboles se pudo confirmar la irrelevancia de una serie de propiedades de características de la señal de la voz extraídas de *openSMILE* debido a que ni siquiera aparecían en dicho árbol. El archivo del documento adjuntado en el [Anexo D](#) corresponde al mismo ejemplo que el de la *Ilustración 14*. De esto se puede observar que sólo 7 de las 31 propiedades aparecen en el árbol, por tanto, 24 de dichas propiedades son redundantes o no aportan suficiente información para la decisión.

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

```

audspec_lengthLlnorm_sma_quartile3 <= 324133.3333
|  audspec_lengthLlnorm_sma_percentile1.0 <= 53779.41176: SU (5.0/2.0)
|  audspec_lengthLlnorm_sma_percentile1.0 > 53779.41176
|  |  audspec_lengthLlnorm_sma_upleveltime75 <= 166975.4926: D (2.0)
|  |  audspec_lengthLlnorm_sma_upleveltime75 > 166975.4926
|  |  |  audspec_lengthLlnorm_sma_quartile2 <= 221500
|  |  |  |  audspec_lengthLlnorm_sma_upleveltime90 <= 166992.0643: SA (4.0)
|  |  |  |  audspec_lengthLlnorm_sma_upleveltime90 > 166992.0643: N (10.0/2.0)
|  |  |  audspec_lengthLlnorm_sma_quartile2 > 221500: SA (5.0/1.0)
audspec_lengthLlnorm_sma_quartile3 > 324133.3333
|  audspec_lengthLlnorm_sma_percentile99.0 <= 1165323.286
|  |  audspec_lengthLlnorm_sma_range <= 222158.0148: D (5.0/1.0)
|  |  audspec_lengthLlnorm_sma_range > 222158.0148
|  |  |  audspec_lengthLlnorm_sma_range <= 290851.1111: SU (4.0)
|  |  |  audspec_lengthLlnorm_sma_range > 290851.1111
|  |  |  |  audspec_lengthLlnorm_sma_range <= 363992.5926: A (2.0)
|  |  |  |  audspec_lengthLlnorm_sma_range > 363992.5926: D (3.0/1.0)
|  audspec_lengthLlnorm_sma_percentile99.0 > 1165323.286: A (5.0)

```

Ilustración 16. Árbol de decisión bloque1_1

Además de poder extraer el árbol de decisión, WEKA nos muestra tanto una matriz de confusión como el porcentaje de fiabilidad y error de dichas características para deducir la clase.

La matriz de confusión hallada del ejemplo es la siguiente:

```

=== Confusion Matrix ===
a b c d e  <-- classified as
7 0 1 1 0 | a = A
0 8 0 0 1 | b = D
0 0 8 0 1 | c = SA
0 2 0 7 0 | d = SU
0 0 0 1 8 | e = N

```

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

Como se puede observar el margen de error es muy pequeño, por lo tanto, las características que se muestran en el árbol de decisión de la *Ilustración 14* son capaces de realizar el reconocimiento de emociones con alta fiabilidad, concretamente con un 84.4444 % de fiabilidad lo que conlleva a un margen de error del 15.5556 %.

Esta extracción del árbol de decisión y porcentaje de fiabilidad y error se realizó con todos los pesos de los bloques de características.

Para concluir qué características son las que aportan mayor fiabilidad se realizó un estudio comparativo de los porcentajes de fiabilidad y error que aportaban cada uno de los pesos de todas las características. De esta manera se puede saber con certeza qué características nos aportan más información a la hora de reconocer emociones. Hay que destacar que con las características del *Bloque3_1* y las del *Bloque1_24* se obtienen los peores resultados, concretamente, una fiabilidad del 20 % y del 35.5556 %, respectivamente. Con lo que se podría suponer que estas características no son útiles para el reconocimiento de emociones ya que su margen de error es demasiado alto.

Dicho estudio se adjunta en el [Anexo E](#).

Una vez realizada la subfase de entrenamiento, en el que se genera un árbol de decisión de cada conjunto de SOMs, se realiza una subfase de reconocimiento de nuevas muestras, para ver la capacidad de clasificación de esta solución. A continuación, se muestra a modo de ejemplo lo que sucede al introducir muestras nuevas en el árbol de decisión correspondiente al conjunto de características del *bloque1_1*.

- Test *angry*: en la siguiente matriz se observa que se obtiene un 100 % de aciertos al introducir una muestra nueva de la emoción *enfado* al árbol de decisión entrenado:

```
=== Confusion Matrix ===
```

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

```
a b c d e <-- classified as
9 0 0 0 0 | a = A
0 0 0 0 0 | b = D
0 0 0 0 0 | c = SA
0 0 0 0 0 | d = SU
0 0 0 0 0 | e = N
```

- **Test *neutral*:** en la siguiente matriz de confusión de la muestra nueva correspondiente a la emoción *neutral* se obtiene un 0 % de aciertos, siendo confundida con la emoción *asco* en un 44.4444 % de las veces.

```
=== Confusion Matrix ===
```

```
a b c d e <-- classified as
0 0 0 0 0 | a = A
0 0 0 0 0 | b = D
0 0 0 0 0 | c = SA
0 0 0 0 0 | d = SU
2 4 1 2 0 | e = N
```

- **Test *surprised*:** la siguiente matriz corresponde a la obtenida a partir de una muestra nueva de a emoción *sorpresas*. Resulta con un 22.2222% de aciertos.

```
=== Confusion Matrix ===
```

```
a b c d e <-- classified as
0 0 0 0 0 | a = A
0 0 0 0 0 | b = D
0 0 0 0 0 | c = SA
5 2 0 2 0 | d = SU
0 0 0 0 0 | e = N
```

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

- Test *sad*: esta matriz corresponde a la muestra de la emoción *tristeza*, obteniéndose una mala clasificación en el 100 % de los casos, es decir, un 0 % de aciertos.

```
=== Confusion Matrix ===  
  
a b c d e  <-- classified as  
0 0 0 0 0 | a = A  
0 0 0 0 0 | b = D  
6 2 0 1 0 | c = SA  
0 0 0 0 0 | d = SU  
0 0 0 0 0 | e = N
```

- Test *disgust*: por último, esta matriz muestra un 44.4444 % de aciertos cuando se introduce una muestra de la emoción *asco* en el árbol entrenado.

```
=== Confusion Matrix ===  
  
a b c d e  <-- classified as  
0 0 0 0 0 | a = A  
0 4 3 1 1 | b = D  
0 0 0 0 0 | c = SA  
0 0 0 0 0 | d = SU  
0 0 0 0 0 | e = N
```

Los resultados obtenidos tras las pruebas, como se ha podido observar, no son muy buenos en general. En el caso de la emoción *enfado* siempre se obtiene una buena clasificación, y en la emoción *asco* la clasificación no es del todo mala. Pero en general es una clasificación pésima, que no coincide con el porcentaje de fiabilidad obtenido al generar los árboles de decisión.

Los resultados de las pruebas del resto de árboles de decisión siguen la misma línea que el del ejemplo mostrado, por tanto, la fiabilidad de utilizar esta técnica con los pesos de las características no es del todo bueno a la hora de reconocer emociones.

7.3.2 FASE REDUCCIÓN DE DATOS

Para seguir con la identificación de las características que más información aportan para el reconocimiento de emociones, se ha reducido el número de características utilizadas a las que tienen una fiabilidad mayor al 90%. De esta manera se identificará mejor qué propiedades de la voz nos están siendo más útiles para el reconocimiento de emociones, eliminando información que sature el método utilizado causando los problemas detectados en la fase anterior.

Esta reducción de los datos de entrada se ha realizado de la siguiente manera: se han recogido las características que aparecen en los árboles de decisión que tienen un porcentaje de fiabilidad mayor al 90%, resumidas en la *Tabla 5*. Con este nuevo conjunto de características se ha repetido el proceso, es decir, se han realizado cinco SOMs nuevos con estas características, extrayendo sus pesos para, a su vez, introducirlos en WEKA con el fin de extraer un único y concluyente árbol de decisión. Dicho árbol se muestra en la *Ilustración 15*.

BLOQUE	CARACTERÍSTICA
Bloque1_3	pcm_RMSenergy_sma_
Bloque1_5	audspec_lengthL1norm_sma_de_
Bloque1_12	audSpec_Rfilt_sma [3]
Bloque1_20	audSpec_Rfilt_sma[11]
Bloque1_25	audSpec_Rfilt_sma[16]
Bloque1_28	audSpec_Rfilt_sma[19]
Bloque1_39	pcm_fftMag_spectralRollOff75.0_sma_
Bloque1_41	pcm_fftMag_spectralFlux_sma_

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

Bloque1_44	pcm_fftMag_spectralVariance_sma_
Bloque1_47	pcm_fftMag_spectralSlope_sma_
Bloque1_49	pcm_fftMag_spectralHarmonicity_sma_
Bloque1_58	mfcc_sma [9]
Bloque1_87	mfcc_sma [23]
Bloque1_91	pcm_fftMag_fband1000-4000_sma_de_
Bloque1_94	pcm_fftMag_spectralRollOff75.0_sma_de_
Bloque1_97	pcm_fftMag_spectralCentroid_sma_de_
Bloque1_104	pcm_fftMag_spectralHarmonicity_sma_de_
Bloque1_117	mfcc_sma_de [13]
Bloque4_3	pcm_RMSenergy_sma_
Bloque4_37	pcm_fftMag_spectralFlux_sma_
Bloque4_51	mfcc_sma [6]

Tabla 5. Características con árbol de decisión fiabilidad > 90 %

```

pcm_fftMag_spectralFlux_sma_quartile2 <= 41079.58737
| pcm_fftMag_fband1000-4000_sma_de_percentile1.0 <= -69.641567: SA (10.0/1.0)
| pcm_fftMag_fband1000-4000_sma_de_percentile1.0 > -69.641567: N (8.0)
pcm_fftMag_spectralFlux_sma_quartile2 > 41079.58737
| audspec_lengthLlnorm_sma_de_range <= 125650
| | audSpec_Rfilt_sma[16]_risetime <= 197647.1647: D (8.0)
| | audSpec_Rfilt_sma[16]_risetime > 197647.1647: SU (10.0/1.0)
| audspec_lengthLlnorm_sma_de_range > 125650: A (9.0)
  
```

Ilustración 17. Árbol de decisión fase reducción de datos

Como se puede observar esta técnica vuelve a realizar una reducción de la información, desapareciendo las propiedades de la voz que menos información aportan y resultado las

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

características que se muestran en la *Tabla 6*. La reducción es considerable, ya que de la cantidad de datos introducidos en un principio resultan únicamente cuatro.

BLOQUE	CARACTERÍSTICA
Bloque1_5_1	audspec_lengthL1norm_sma_de_range
Bloque1_25_24	audSpec_Rfilt_sma[16]_risetime
Bloque1_41_5	pcm_fftMag_spectralFlux_sma_quartile2
Bloque1_91_10	pcm_fftMag_fband1000-4000_sma_de_percentile1.0

Tabla 6. Propiedades de la voz resultantes SOM con Árbol de decisión fase 2

Otro dato importante que destacar es que estos resultados aportan un porcentaje de fiabilidad al reconocimiento de emociones de un 95.556 % con un margen de error del 4.4444 %, lo cual quiere decir que mejora muchos resultados del estudio de fiabilidad realizado y mostrado en el [Anexo E](#), a excepción de algunos conjuntos de características destacando el *Bloque1_94* que lo supera con un 97.7778 %. A pesar de ello, el obtener una fiabilidad de casi 96 % con tan sólo 4 propiedades de la voz es mucho más ventajoso que realizarlo con una mayor cantidad de datos como es el del conjunto del *Bloque1_94*.

La siguiente matriz de confusión muestra más detalladamente qué emociones tienen mayor posibilidad de ser mal clasificadas al realizarlo con estas propiedades. Como se puede observar, el error sólo aparece en un 12.5 % de las muestras de *asco*, ocurriendo lo mismo con las de la emoción *neutral*. A pesar de ello, el resultado es muy bueno y se confirmará a través de las pruebas realizadas sobre muestras nuevas que se muestran a continuación.

```

=== Confusion Matrix ===
a b c d e <-- classified as
9 0 0 0 0 | a = A
  
```

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

```
0 8 0 1 0 | b = D
0 0 9 0 0 | c = SA
0 0 0 9 0 | d = SU
0 0 1 0 8 | e = N
```

Las pruebas realizadas consisten en utilizar el árbol de decisión mostrado en la *Ilustración 15*, es decir, con las características de los árboles con una fiabilidad mayor al 90 %, obteniéndose los siguientes resultados:

- Test *angry*: en primer lugar, se ha testeado una muestra nueva de la emoción *enfado*. Los resultados, traducidos una vez más en una matriz de confusión, muestran un 77.7778 % de acierto.

```
==== Confusion Matrix ====

a b c d e  <-- classified as
7 0 0 0 2 | a = A
0 0 0 0 0 | b = D
0 0 0 0 0 | c = SA
0 0 0 0 0 | d = SU
0 0 0 0 0 | e = N
```

- Test *disgust*: en el caso de introducir una muestra nueva de la emoción *asco* en el árbol de decisión entrenado, no se produce ningún acierto en la clasificación de la nueva muestra.

```
==== Confusion Matrix ====

a b c d e  <-- classified as
0 0 0 0 0 | a = A
0 0 1 6 2 | b = D
0 0 0 0 0 | c = SA
0 0 0 0 0 | d = SU
0 0 0 0 0 | e = N
```

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

- Test *neutral*: la siguiente matriz, correspondiente al resultado de introducir una muestra nueva de la emoción *neutral*, presenta un 22.2222 % de acierto en esta clasificación.

```
=== Confusion Matrix ===  
  
a b c d e  <-- classified as  
0 0 0 0 0 | a = A  
0 0 0 0 0 | b = D  
0 0 0 0 0 | c = SA  
0 0 0 0 0 | d = SU  
0 2 0 5 2 | e = N
```

- Test *sad*: la emoción *tristeza*, como ocurre en el caso de la emoción *asco*, presenta un 0 % de aciertos en muestras nuevas.

```
=== Confusion Matrix ===  
  
a b c d e  <-- classified as  
0 0 0 0 0 | a = A  
0 0 0 0 0 | b = D  
6 1 0 2 0 | c = SA  
0 0 0 0 0 | d = SU  
0 0 0 0 0 | e = N
```

- Test *surprised*: por último, la muestra de *sorpresa* aparece clasificada con un 33.3333% de aciertos para muestras nuevas.

```
=== Confusion Matrix ===  
  
a b c d e  <-- classified as  
0 0 0 0 0 | a = A
```

SOLUCIÓN SOM JUNTO A ÁRBOL DE DECISIÓN

```
0 0 0 0 0 | b = D
0 0 0 0 0 | c = SA
4 2 0 3 0 | d = SU
0 0 0 0 0 | e = N
```

Como se ha podido observar, la emoción *enfado* sigue teniendo una buena clasificación con esta técnica y estas características. La emoción *neutral*, aunque no siga clasificándose bien por lo menos tiene un 22.2222% de aciertos en este caso. La emoción *sorpresa* también mejora, aunque no de una manera significativa. La emoción *tristeza* se mantiene en un 0 % de aciertos. Y, por último, la emoción *asco* mejora considerablemente llegando a un 44.4444 % de aciertos.

En resumidas cuentas, esta reducción de los datos no ha sido especialmente útil a la hora de encontrar las características más importantes para obtener una clasificación de las emociones, aunque la integración del árbol de decisión en el SOM sí que mejora los resultados obtenidos, sobre todo, en los de la emoción *enfado*.

Capítulo 8. CONCLUSIONES Y TRABAJOS FUTUROS

Los objetivos iniciales de este proyecto eran dos. Por un lado, la detección de las características más relevantes para la detección de emociones en la voz, y por otro encontrar cuál era el mejor método de los tres utilizados para el reconocimiento de emociones en la voz.

En el proyecto se han manejado un total de 6373 características. Para poder detectar cuáles son las más relevantes, tras las pruebas de cada solución se intentaba extraer las mismas de los resultados obtenidos:

- La solución EmoVoice es de aprendizaje supervisado. Aunque obtiene los mejores resultados clasificatorios ante muestras nuevas, no provee de información sobre las características mejores para poder hacer dicha clasificación.
- Un SOM intenta aprender (memorizar) una emoción, extrayendo las características más relevantes de la misma. Se desconoce si ha sido por el tamaño de la base de datos de pruebas (tanto por sobreentrenamiento como escasez del mismo) o por la división que se ha hecho de categorías en bloques (y no tener suficientes neuronas para aprender todas las categorías al tiempo). Este segundo posible problema se podría trabajar con un nuevo planteamiento, en el que no se pretendiera tanto la detección de las características relevantes, y se trabajara en un mejor entrenamiento de una red completa.
- Aunque un árbol C4.5 es *a priori* un buen sistema para la clasificación, que permitiría detectar las mejores propiedades para la misma, tiene problemas de sobreajuste y de encontrar soluciones localmente buenas, no necesariamente óptimas desde un punto de vista global. Esto es lo que aparentemente nos hemos encontrado en las pruebas

CONCLUSIONES Y TRABAJOS FUTUROS

realizadas, que en algunos casos se han reducido demasiado los árboles, hasta el punto de que posteriormente no se clasificaban bien las muestras nuevas. En cualquier caso, un árbol C4.5 mejora sustancialmente el resultado de la solución sólo con SOM cuando lo complementa.

Sería interesante, aparte de realizar los trabajos indicados sobre SOM, que emula un sistema de memoria, realizar pruebas adicionales con algún método de redes neuronales supervisado, como puede ser un perceptrón multicapa, que es un tipo de red ampliamente utilizado en clasificación de patrones. El principal problema es que este tipo de red servirá para previsiblemente realizar una mejor clasificación de patrones, pero no permitirá averiguar las características más relevantes, por la excesiva complejidad de la propia red.

Para ampliar el trabajo sobre cuáles son las mejores características para la detección de emociones en la voz, sería conveniente ampliar el estado del arte del mismo desde un punto de vista lingüístico. Aunque inicialmente se habían considerado cinco emociones básicas en la voz, existen diversas opiniones al respecto, por lo que algunas de las aquí consideradas podrían ser más complejas de lo que inicialmente se pensaba.

Capítulo 9. BIBLIOGRAFÍA

- [5] <https://dialnet.unirioja.es/servlet/articulo?codigo=66004>
- [6] https://harmonia.la/mente/cientificos_investigacion_como_tu_voz_refleja_tus_emociones
- [7] <https://neuromarketing.la/2017/10/analisis-de-emociones-a-traves-de-la-voz/>
- [8] https://www.tecnonews.info/noticias/reconocedor_de_voz_que_integra_la_deteccion_de_emociones
- [9] <https://iptc.upm.es/sistema-de-reconocimiento-de-emociones-partir-de-la-voz>
- [10] <https://www.movilonia.com/noticias/aplicacion-reconocimiento-facial/>
- [11] https://es.wikipedia.org/wiki/Se%C3%B1al_de_voz
- [12] <http://www.redalyc.org/html/2010/201017296007/>
- [13] https://es.wikipedia.org/wiki/Red_neuronal_artificial
- [14] <https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-están-volviendo>
- [15] https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte
- [16] <https://web.archive.org/web/20140801145654/http://www.gts.tsc.uvigo.es/~jalba/doctorado/SVM.pdf>
- [17] <file:///C:/Users/usuario/Downloads/Dialnet-LASMAQUINASDESOPORTEVECTORIALSVMS-4838384.pdf>
- [18] <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/6895>

BIBLIOGRAFÍA

- [19] <https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-diagrama-de-%C3%A1rbol-de-decisi%C3%B3n>
- [20] <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf>
- [21] <https://audeering.com/technology/opensmile/>
- [22] <https://rawgit.com/hcmlab/emovoice/master/docs/index.html>
- [23] https://es.wikipedia.org/wiki/Mapa_autoorganizado
- [24] <https://www.tesisenred.net/bitstream/handle/10803/9441/tjjmm1de1.pdf?sequence=1>
- [25] https://es.wikipedia.org/wiki/%C3%81rbol_de_decisi%C3%B3n
- [26] <https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-diagrama-de-%C3%A1rbol-de-decisi%C3%B3n>
- [27] <http://www.elmundo.es/economia/2015/07/31/55bb4d4322601d59248b4582.html>
- [28] <https://smartlaboratory.org/ravdess/>
- [29] <https://es.wikipedia.org/wiki/C4.5>
- [30] http://www.konradlorenz.edu.co/images/stories/suma_digital_sistemas/2009_01/final_paula_andrea.pdf

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

ANEXO A: ARCHIVO DE CONFIGURACIÓN

COMPARE.CONF

```
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
//////////////////////////////////// > openSMILE configuration file for ComParE <  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
//////////////////////////////////// (c) audEERING UG (haftungsbeschränkt),  
////////////////////////////////////  
//////////////////////////////////// All rights reserved.  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
  
;;;;;;;;; component list ;;;;;;;;;;  
  
[componentInstances:cComponentManager]  
instance[dataMemory].type=cDataMemory  
printLevelStats=0  
  
;;;;;;;;; main section ;;;;;;;;;;  
  
;[componentInstances:cComponentManager]  
;instance[ssiSource].type = cSSISource  
  
// --- SSI Source ---  
// the component name must be provided to SSI via the XML-File  
;[ssiSource:cSSISource]  
;writer.dmLevel = wave  
// --- End SSI Source ---  
  
;;;;;;;;;
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
//Componente por mi propio source:
instance[waveSource].type = cWaveSource

;;; default (template) configuration section for component 'cWaveSource'
;;;

[waveSource:cWaveSource]

writer.dmLevel = wave

period = 0

;filename = input.wav

filename = \cm[inputfile(I){input.wav};file name of the input wave file]

monoMixdown = 1

[componentInstances:cComponentManager]
instance[frame60].type=cFramer
instance[win60].type=cWindower
instance[fft60].type=cTransformFFT
instance[fftmp60].type=cFFTMagphase

[frame60:cFramer]
reader.dmLevel=wave
writer.dmLevel=frame60
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left

[win60:cWindower]
reader.dmLevel=frame60
writer.dmLevel=winG60
winFunc=gauss
gain=1.0
sigma=0.4
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
[fft60:cTransformFFT]
reader.dmLevel=winG60
writer.dmLevel=fftcG60

[fftmp60:cFFTMagphase]
reader.dmLevel=fftcG60
writer.dmLevel=fftmagG60

;;;;;;;;;;;;;

[componentInstances:cComponentManager]
instance[frame25].type=cFramer
;instance[pe25].type=cVectorPreemphasis
instance[win25].type=cWindower
instance[fft25].type=cTransformFFT
instance[fftmp25].type=cFFTMagphase

[frame25:cFramer]
reader.dmLevel=wave
writer.dmLevel=frame25
frameSize = 0.020
frameStep = 0.010
frameCenterSpecial = left

[pe25:cVectorPreemphasis]
reader.dmLevel=frame25
writer.dmLevel=frame25pe
k=0.97
de=0

[win25:cWindower]
reader.dmLevel=frame25
writer.dmLevel=winH25
winFunc=hamming

[fft25:cTransformFFT]
reader.dmLevel=winH25
writer.dmLevel=fftcH25

[fftmp25:cFFTMagphase]
reader.dmLevel=fftcH25
writer.dmLevel=fftmagH25
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
;;;;;;;;;;;;; HPS pitch

[componentInstances:cComponentManager]
instance[scale].type=cSpecScale
instance[shs].type=cPitchShs

[scale:cSpecScale]
reader.dmLevel=fftmagG60
writer.dmLevel=hpsG60
copyInputName = 1
processArrayFields = 0
scale=octave
sourceScale = lin
interpMethod = spline
minF = 25
maxF = -1
nPointsTarget = 0
specSmooth = 1
specEnhance = 1
auditoryWeighting = 1

[shs:cPitchShs]
reader.dmLevel=hpsG60
writer.dmLevel=pitchShsG60
writer.levelconf.nT=100
copyInputName = 1
processArrayFields = 0
maxPitch = 620
minPitch = 52
nCandidates = 6
scores = 1
voicing = 1
FOC1 = 0
voicingC1 = 0
FOraw = 1
voicingClip = 1
voicingCutoff = 0.700000
inputFieldSearch = Mag_octScale
octaveCorrection = 0
nHarmonics = 15
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
compressionFactor = 0.850000
greedyPeakAlgo = 1

;;;;; Pitch with Viterbi smoother
[componentInstances:cComponentManager]
instance[energy60].type=cEnergy

[energy60:cEnergy]
reader.dmLevel=winG60
writer.dmLevel=e60
; This must be > than buffersize of viterbi smoother
writer.levelconf.nT=100
rms=1
log=0

[componentInstances:cComponentManager]
instance[pitchSmoothViterbi].type=cPitchSmootherViterbi

[pitchSmoothViterbi:cPitchSmootherViterbi]
reader.dmLevel=pitchShsG60
reader2.dmLevel=pitchShsG60
writer.dmLevel=pitchG60_viterbi
copyInputName = 1
bufferLength=30
F0final = 1
F0finalEnv = 0
voicingFinalClipped = 0
voicingFinalUnclipped = 1
F0raw = 0
voicingCl = 0
voicingClip = 0
wTvv =10.0
wTvvd= 5.0
wTvuv=10.0
wThr = 4.0
wTuu = 0.0
wLocal=2.0
wRange=1.0

[componentInstances:cComponentManager]
instance[volmerge].type = cValbasedSelector

[volmerge:cValbasedSelector]
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
reader.dmLevel = e60;pitchG60_viterbi
writer.dmLevel = pitchG60
idx=0
threshold=0.001
removeIdx=1
zeroVec=1
outputVal=0.0

;;;;;;;;;;;;; Voice Quality (VQ)

[componentInstances:cComponentManager]
instance[pitchJitter].type=cPitchJitter

[pitchJitter:cPitchJitter]
reader.dmLevel = wave
writer.dmLevel = jitterShimmer
copyInputName = 1
F0reader.dmLevel = pitchG60
F0field = F0final
searchRangeRel = 0.250000
jitterLocal = 1
jitterDDP = 1
jitterLocalEnv = 0
jitterDDPEnv = 0
shimmerLocal = 1
shimmerLocalEnv = 0
onlyVoiced = 0
logHNR = 1
inputMaxDelaySec = 2.0
;periodLengths = 0
;periodStarts = 0

;;;;;;;;;;;;; Energy / loudness

[componentInstances:cComponentManager]
instance[energy].type=cEnergy
instance[melspec1].type=cMelspec
instance[audspec].type=cPlp
instance[audspecRasta].type=cPlp
instance[audspecSum].type=cVectorOperation
instance[audspecRastaSum].type=cVectorOperation
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
[energy:cEnergy]
reader.dmLevel = frame25
writer.dmLevel = energy
log=0
rms=1

[melspec1:cMelspec]
reader.dmLevel=fftmagH25
writer.dmLevel=melspec1
; htk compatible sample value scaling
htkcompatible = 0
nBands = 26
; use power spectrum instead of magnitude spectrum
usePower = 1
lofreq = 20
hifreq = 8000
specScale = mel
showFbank = 0

; perform auditory weighting of spectrum
[audspec:cPlp]
reader.dmLevel=melspec1
writer.dmLevel=audspec
firstCC = 0
lpOrder = 5
cepLifter = 22
compression = 0.33
htkcompatible = 0
doIDFT = 0
doLpToCeps = 0
doLP = 0
doInvLog = 0
doAud = 1
doLog = 0
newRASTA=0
RASTA=0

; perform RASTA style filtering of auditory spectra
[audspecRasta:cPlp]
reader.dmLevel=melspec1
writer.dmLevel=audspecRasta
nameAppend = Rfilt
firstCC = 0
```


ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
lpOrder = 5
cepLifter = 22
compression = 0.33
htkcompatible = 0
doIDFT = 0
doLpToCeps = 0
doLP = 0
doInvLog = 0
doAud = 1
doLog = 0
newRASTA=1
RASTA=0

[audspecSum:cVectorOperation]
reader.dmLevel = audspec
writer.dmLevel = audspecSum
// nameAppend =
copyInputName = 1
processArrayFields = 0
operation = 111
nameBase = audspec

[audspecRastaSum:cVectorOperation]
reader.dmLevel = audspecRasta
writer.dmLevel = audspecRastaSum
// nameAppend =
copyInputName = 1
processArrayFields = 0
operation = 111
nameBase = audspecRasta

;;;;;;;;;;;;; spectral

[componentInstances:cComponentManager]
instance[spectral].type=cSpectral

[spectral:cSpectral]
reader.dmLevel=fftmagH25
writer.dmLevel=spectral
bands[0]=250-650
bands[1]=1000-4000
rollOff[0] = 0.25
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
rollOff[1] = 0.50
rollOff[2] = 0.75
rollOff[3] = 0.90
flux=1
centroid=1
maxPos=0
minPos=0
entropy=1
variance=1
skewness=1
kurtosis=1
slope=1
harmonicity=1
sharpness=1

;;;;;;;;;;;;; mfcc

[componentInstances:cComponentManager]
instance[melspecMfcc].type=cMelspec
instance[mfcc].type=cMfcc

[melspecMfcc:cMelspec]
reader.dmLevel=fftmagH25
writer.dmLevel=melspecMfcc
copyInputName = 1
processArrayFields = 1
; htk compatible sample value scaling
htkcompatible = 1
nBands = 26
; use power spectrum instead of magnitude spectrum
usePower = 1
lofreq = 20
hifreq = 8000
specScale = mel
inverse = 0

[mfcc:cMfcc]
reader.dmLevel=melspecMfcc
writer.dmLevel=mfcc1_12
copyInputName = 0
processArrayFields = 1
firstMfcc = 1
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
lastMfcc = 14
cepLifter = 22.0
htkcompatible = 1

;;;;;;;;;;;;; zcr

[componentInstances:cComponentManager]
instance[mzcr].type=cMZcr

[mzcr:cMZcr]
reader.dmLevel = frame60
writer.dmLevel = zcr
copyInputName = 1
processArrayFields = 1
zcr = 1
mcr = 0
amax = 0
maxmin = 0
dc = 0

;;;;;;;;;;;;; smoothing

[componentInstances:cComponentManager]
instance[smoNz].type=cContourSmoother
instance[smoA].type=cContourSmoother
instance[smoB].type=cContourSmoother
instance[f0sel].type=cDataSelector

[smoNz:cContourSmoother]
reader.dmLevel = pitchG60;jitterShimmer
writer.dmLevel = lld_nzsmo
writer.levelconf.isRb=0
writer.levelconf.growDyn=1
nameAppend = sma
copyInputName = 1
noPostEOIprocessing = 0
smaWin = 3
noZeroSma = 1

[f0sel:cDataSelector]
reader.dmLevel = lld_nzsmo
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
writer.dmLevel = lld_f0_nzsma
writer.levelconf.isRb=0
writer.levelconf.growDyn=1
nameAppend = ff0
selected = F0final_sma

[smoA:cContourSmoother]
reader.dmLevel = audspecSum;audspecRastaSum;energy;zcr
writer.dmLevel = lldA_smo
writer.levelconf.isRb=0
writer.levelconf.growDyn=1
nameAppend = sma
copyInputName = 1
noPostEOIprocessing = 0
smaWin = 3

[smoB:cContourSmoother]
reader.dmLevel = audspecRasta;spectral;mfcc1_12
writer.dmLevel = lldB_smo
writer.levelconf.isRb=0
writer.levelconf.growDyn=1
nameAppend = sma
copyInputName = 1
noPostEOIprocessing = 0
smaWin = 3

;;;;;;;;; deltas
[componentInstances:cComponentManager]
instance[deNz].type=cDeltaRegression
instance[deA].type=cDeltaRegression
instance[deB].type=cDeltaRegression
instance[def0sel].type=cDeltaRegression

[deNz:cDeltaRegression]
reader.dmLevel = lld_nzsma
writer.dmLevel = lld_nzsma_de
writer.levelconf.isRb=0
writer.levelconf.growDyn=1
onlyInSegments = 1
zeroSegBound = 1

[deA:cDeltaRegression]
reader.dmLevel = lldA_smo
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
writer.dmLevel = lldA_smo_de
writer.levelconf.isRb=0
writer.levelconf.growDyn=1

[deB:cDeltaRegression]
reader.dmLevel = lldB_smo
writer.dmLevel = lldB_smo_de
writer.levelconf.isRb=0
writer.levelconf.growDyn=1

[def0sel:cDeltaRegression]
reader.dmLevel = lld_f0_nzsmo
writer.dmLevel = lld_f0_nzsmo_de
writer.levelconf.isRb=0
writer.levelconf.growDyn=1
onlyInSegments = 1
zeroSegBound = 1

;;;;;;;;;; functionals / statistics

[componentInstances:cComponentManager]
instance[functionalsA].type=cFunctionals
instance[functionalsB].type=cFunctionals
instance[functionalsF0].type=cFunctionals
instance[functionalsNz].type=cFunctionals

; shared functionals for LLD
instance[functionalsLLD].type=cFunctionals

; shared functionals for Delta LLD
instance[functionalsDelta].type=cFunctionals

; functionals for energy related lld
[functionalsA:cFunctionals]
reader.dmLevel = lldA_smo;lldA_smo_de
writer.dmLevel = functionalsA
// nameAppend =
copyInputName = 1
//frameMode = full
//frameSize = 0
//frameStep = 0
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
//frameCenterSpecial = left
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left

functionalsEnabled = Extremes ; Percentiles ; Moments ; Segments ; Times
; Lpc

Extremes.max = 0
Extremes.min = 0
Extremes.maxpos = 1
Extremes.minpos = 1
Extremes.maxameandist = 0
Extremes.minameandist = 0

Segments.maxNumSeg = 100
Segments.segmentationAlgorithm = relTh
Segments.thresholds = 0.25 ; 0.75
Segments.ravgLng = 3
Segments.numSegments = 0
Segments.meanSegLen = 1
Segments.maxSegLen = 1
Segments.minSegLen = 1
Segments.segLenStddev = 1
Segments.norm = second

Moments.variance = 0
Moments.stddev = 1
Moments.skewness = 1
Moments.kurtosis = 1
Moments.amean = 0

Percentiles.quartiles = 1
Percentiles.iqr = 1
Percentiles.percentile[0] = 0.01
Percentiles.percentile[1] = 0.99
Percentiles.pctlrange[0] = 0-1
Percentiles.interp = 1

Times.upleveltime25 = 1
Times.downleveltime25 = 0
Times.upleveltime50 = 1
Times.downleveltime50 = 0
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
Times.upleveltime75 = 1
Times.downleveltime75 = 0
Times.upleveltime90 = 1
Times.downleveltime90 = 0
Times.risetime = 1
Times.falltime = 0
Times.leftctime = 1
Times.rightctime = 0
Times.duration = 0
Times.buggySecNorm = 0
Times.norm = segment

Lpc.lpGain = 1
Lpc.lpc = 1
Lpc.firstCoeff = 0
Lpc.order = 5

nonZeroFuncs = 0
masterTimeNorm = segment

; functionals for spectrum related lld
[functionalsB:cFunctionals]
reader.dmLevel = lldB_smo;lldB_smo_de
writer.dmLevel = functionalsB
// nameAppend =
copyInputName = 1
//frameMode = full
//frameSize = 0
//frameStep = 0
//frameCenterSpecial = left
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left

functionalsEnabled = Extremes ; Percentiles ; Moments ; Segments ; Times
; Lpc

Extremes.max = 0
Extremes.min = 0
Extremes.maxpos = 1
Extremes.minpos = 1
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
Extremes.maxameandist = 0
Extremes.minameandist = 0

Segments.maxNumSeg = 100
Segments.segmentationAlgorithm = relTh
Segments.thresholds = 0.25 ; 0.75
Segments.rangeRelThreshold = 0.200000
Segments.numSegments = 0
Segments.meanSegLen = 1
Segments.maxSegLen = 1
Segments.minSegLen = 1
Segments.segLenStddev = 1
Segments.norm = second

Moments.variance = 0
Moments.stddev = 1
Moments.skewness = 1
Moments.kurtosis = 1
Moments.amean = 0

Percentiles.quartiles = 1
Percentiles.iqr = 1
Percentiles.percentile[0] = 0.01
Percentiles.percentile[1] = 0.99
Percentiles.pctlrange[0] = 0-1
Percentiles.interp = 1

Times.upleveltime25 = 1
Times.downleveltime25 = 0
Times.upleveltime50 = 1
Times.downleveltime50 = 0
Times.upleveltime75 = 1
Times.downleveltime75 = 0
Times.upleveltime90 = 1
Times.downleveltime90 = 0
Times.risetime = 1
Times.falltime = 0
Times.leftctime = 1
Times.rightctime = 0
Times.duration = 0
Times.buggySecNorm = 0
Times.norm = segment
```


ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
Lpc.lpGain = 1
Lpc.lpc = 1
Lpc.firstCoeff = 0
Lpc.order = 5

nonZeroFuncs = 0
masterTimeNorm = segment

; functionals for pitch onsets/offsets
[functionalsF0:cFunctionals]
reader.dmLevel = lld_f0_nzsmo
writer.dmLevel = functionalsF0
//nameAppend = ff0
copyInputName = 1
//frameMode = full
//frameSize = 0
//frameStep = 0
//frameCenterSpecial = left
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left

functionalsEnabled = Means ; Segments

Means.amean = 0
Means.absmean = 0
Means.qmean = 0
Means.nzamean = 0
Means.nzabsmean = 0
Means.nzqmean = 0
Means.nzgmean = 0
Means.nnz = 1
Means.norm = segment

Segments.maxNumSeg = 100
Segments.segmentationAlgorithm = nonX
Segments.X = 0.0
Segments.numSegments = 0
Segments.meanSegLen = 1
Segments.maxSegLen = 1
Segments.minSegLen = 1
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
Segments.segLenStddev = 1
Segments.norm = second

nonZeroFuncTs = 0
masterTimeNorm = segment

; functionals for pitch and vq related lld in voiced regions
[functionalsNz:cFunctionals]
reader.dmLevel = lld_nzsmo;lld_nzsmo_de
writer.dmLevel = functionalsNz
// nameAppend =
copyInputName = 1
//frameMode = full
//frameSize = 0
//frameStep = 0
//frameCenterSpecial = left
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left
functionalsEnabled = Means ; Extremes ; Regression ; Percentiles ;
Moments ; Times ; Lpc
Means.amean = 1
Means.posamean = 1
Means.absmean = 0
Means.qmean = 0
Means.rqmean = 1
Means.nzamean = 0
Means.nzabsmean = 0
Means.nzqmean = 0
Means.posrqmean = 0
Means.nzgmean = 0
Means.nnz = 0
Means.flatness = 1
Means.norm = frames
Extremes.max = 0
Extremes.min = 0
Extremes.maxpos = 1
Extremes.minpos = 1
Extremes.maxameandist = 0
Extremes.minameandist = 0
Moments.variance = 0
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
Moments.stddev = 1
Moments.skewness = 1
Moments.kurtosis = 1
Moments.amean = 0
Regression.linregc1 = 1
Regression.linregc2 = 1
Regression.linregerrA = 0
Regression.linregerrQ = 1
Regression.qregc1 = 1
Regression.qregc2 = 1
Regression.qregc3 = 1
Regression.qregerrA = 0
Regression.qregerrQ = 1
Regression.oldBuggyQerr = 0
Regression.centroid = 1
Regression.normRegCoeff = 0
Percentiles.quartiles = 1
Percentiles.iqr = 1
Percentiles.percentile[0] = 0.01
Percentiles.percentile[1] = 0.99
Percentiles.pctlrange[0] = 0-1
Percentiles.interp = 1
Times.upleveltime25 = 1
Times.downleveltime25 = 0
Times.upleveltime50 = 1
Times.downleveltime50 = 0
Times.upleveltime75 = 1
Times.downleveltime75 = 0
Times.upleveltime90 = 1
Times.downleveltime90 = 0
Times.risetime = 1
Times.falltime = 0
Times.leftctime = 1
Times.rightctime = 0
Times.duration = 0
Times.buggySecNorm = 0
Times.norm = segment
Lpc.lpGain = 1
Lpc.lpc = 1
Lpc.firstCoeff = 0
Lpc.order = 5
nonZeroFuncs = 1
masterTimeNorm = segment
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
[functionalsLLD:cFunctionals]
reader.dmLevel = lldA_smo;lldB_smo
writer.dmLevel = functionalsLLD

copyInputName = 1
//frameMode = full
//frameSize = 0
//frameStep = 0
//frameCenterSpecial = left
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left

functionalsEnabled = Means ; Peaks2 ; Regression

Means.amean = 1
Means.posamean = 0
Means.absmean = 0
Means.qmean = 0
Means.rqmean = 1
Means.nzamean = 0
Means.nzabsmean = 0
Means.nzqmean = 0
Means.posrqmean = 0
Means.nzqmean = 0
Means.nnz = 0
Means.flatness = 1
Means.norm = frames

Regression.linregc1 = 1
Regression.linregc2 = 1
Regression.linregerrA = 0
Regression.linregerrQ = 1
Regression.qregc1 = 1
Regression.qregc2 = 1
Regression.qregc3 = 1
Regression.qregerrA = 0
Regression.qregerrQ = 1
Regression.oldBuggyQerr = 0
Regression.centroid = 1
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
Regression.normRegCoeff = 0

Peaks2.numPeaks = 0
Peaks2.meanPeakDist = 1
Peaks2.meanPeakDistDelta = 0
Peaks2.peakDistStddev = 1
Peaks2.peakRangeAbs = 1
Peaks2.peakRangeRel = 1
Peaks2.peakMeanAbs = 1
Peaks2.peakMeanMeanDist = 1
Peaks2.peakMeanRel = 1
Peaks2.ptpAmpMeanAbs = 0
Peaks2.ptpAmpMeanRel = 0
Peaks2.ptpAmpStddevAbs = 0
Peaks2.ptpAmpStddevRel = 0
Peaks2.minRangeAbs = 0
Peaks2.minRangeRel = 1
Peaks2.minMeanAbs = 0
Peaks2.minMeanMeanDist = 0
Peaks2.minMeanRel = 0
Peaks2.mtmAmpMeanAbs = 0
Peaks2.mtmAmpMeanRel = 0
Peaks2.mtmAmpStddevAbs = 0
Peaks2.mtmAmpStddevRel = 0
Peaks2.meanRisingSlope = 1
Peaks2.maxRisingSlope = 0
Peaks2.minRisingSlope = 0
Peaks2.stddevRisingSlope = 1
Peaks2.meanFallingSlope = 1
Peaks2.maxFallingSlope = 0
Peaks2.minFallingSlope = 0
Peaks2.stddevFallingSlope = 1
Peaks2.norm = seconds
Peaks2.relThresh = 0.100000
Peaks2.dynRelThresh = 0
;Peaks2.posDbgOutp = minmax.txt
Peaks2.posDbgAppend = 0
Peaks2.consoleDbg = 0

[functionalsDelta:cFunctionals]
reader.dmLevel = lldA_smo_de;lldB_smo_de
writer.dmLevel = functionalsDelta
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
copyInputName = 1
//frameMode = full
//frameSize = 0
//frameStep = 0
//frameCenterSpecial = left
frameSize = 0.060
frameStep = 0.010
frameCenterSpecial = left

functionalsEnabled = Means ; Peaks2

Means.amean = 0
Means.posamean = 1
Means.absmean = 0
Means.qmean = 0
Means.rqmean = 1
Means.nzamean = 0
Means.nzabsmean = 0
Means.nzqmean = 0
Means.posrqmean = 0
Means.nzgmean = 0
Means.nnz = 0
Means.flatness = 1
Means.norm = frames

Peaks2.numPeaks = 0
Peaks2.meanPeakDist = 1
Peaks2.meanPeakDistDelta = 0
Peaks2.peakDistStddev = 1
Peaks2.peakRangeAbs = 1
Peaks2.peakRangeRel = 1
Peaks2.peakMeanAbs = 1
Peaks2.peakMeanMeanDist = 1
Peaks2.peakMeanRel = 1
Peaks2.ptpAmpMeanAbs = 0
Peaks2.ptpAmpMeanRel = 0
Peaks2.ptpAmpStddevAbs = 0
Peaks2.ptpAmpStddevRel = 0
Peaks2.minRangeAbs = 0
Peaks2.minRangeRel = 1
Peaks2.minMeanAbs = 0
Peaks2.minMeanMeanDist = 0
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
Peaks2.minMeanRel = 0
Peaks2.mtmAmpMeanAbs = 0
Peaks2.mtmAmpMeanRel = 0
Peaks2.mtmAmpStddevAbs = 0
Peaks2.mtmAmpStddevRel = 0
Peaks2.meanRisingSlope = 1
Peaks2.maxRisingSlope = 0
Peaks2.minRisingSlope = 0
Peaks2.stddevRisingSlope = 1
Peaks2.meanFallingSlope = 1
Peaks2.maxFallingSlope = 0
Peaks2.minFallingSlope = 0
Peaks2.stddevFallingSlope = 1
Peaks2.norm = seconds
Peaks2.relThresh = 0.100000
Peaks2.dynRelThresh = 0
;Peaks2.posDbgOutp = minmax.txt
Peaks2.posDbgAppend = 0
Peaks2.consoleDbg = 0

;;;;;;;;;; prepare features for standard output module

[componentInstances:cComponentManager]
instance[lldconcat].type=cVectorConcat
instance[llddeconcat].type=cVectorConcat
instance[funconcat].type=cVectorConcat

[lldconcat:cVectorConcat]
reader.dmLevel = lld_nzsmo;lldA_smo;lldB_smo
writer.dmLevel = lld
includeSingleElementFields = 1

[llddeconcat:cVectorConcat]
reader.dmLevel = lld_nzsmo_de;lldA_smo_de;lldB_smo_de
writer.dmLevel = lld_de
includeSingleElementFields = 1

[funconcat:cVectorConcat]
reader.dmLevel =
functionalsA;functionalsB;functionalsNz;functionalsF0;functionalsLLD;func
tionalsDelta
```

ANEXO A: ARCHIVO DE CONFIGURACIÓN COMPARE.CONF

```
writer.dmLevel = func
includeSingleElementFields = 1

;;;;;;;;;;;;; csv output of functionals and temporal descriptors
;;;;;;;;;;;;;

[componentInstances:cComponentManager]
instance[lld].type=cCsvSink

[lld:cCsvSink]
reader.dmLevel=func
;filename=test.csv

filename = \cm[outputfile(0){output.csv}:file name of the output CSV
file]
append = 1
timestamp = 1
number = 0
printHeader = 1

;;;;;;;;;;;;; binary output of functionals and temporal
descriptors ;;;;;;;;;;;;;;
;[componentInstances:cComponentManager]
;instance[ssiSink].type = cSSISink

// --- SSI-Sink ---
// the component name must be provided to SSI via the XML-File
;[ssiSink:cSSISink]
;reader.dmLevel = func
// --- End SSI-Sink ---
```


ANEXO B: CARACTERÍSTICAS OPENSMILE

Este anexo contiene las características que extrae *openSMILE* de un audio a través del archivo de configuración *compare.conf* ([Anexo A](#)).

La primera columna indica la enumeración de las características de cada bloque, la segunda columna contiene las características del bloque al que pertenecen en la tercera columna y, por último, en la cuarta columna, se incluyen las propiedades o subcaracterísticas de cada una de las características extraídas por *openSMILE*. Por ejemplo, la característica 1 del Bloque 1 en realidad son las siguientes 31 (sub)características:

1. audspec_lengthL1norm_sma_range
2. audspec_lengthL1norm_sma_maxPos
3. audspec_lengthL1norm_sma_minPos
4. audspec_lengthL1norm_sma_quartile1
5. audspec_lengthL1norm_sma_quartile2
6. audspec_lengthL1norm_sma_quartile3
7. audspec_lengthL1norm_sma_iqr1-2
8. audspec_lengthL1norm_sma_iqr2-3
9. audspec_lengthL1norm_sma_iqr1-3
10. audspec_lengthL1norm_sma_percentile1.0
11. audspec_lengthL1norm_sma_percentile99.0
12. audspec_lengthL1norm_sma_pctlrange0-1
13. audspec_lengthL1norm_sma_stddev
14. audspec_lengthL1norm_sma_skewness
15. audspec_lengthL1norm_sma_kurtosis
16. audspec_lengthL1norm_sma_meanSegLen
17. audspec_lengthL1norm_sma_maxSegLen
18. audspec_lengthL1norm_sma_minSegLen
19. audspec_lengthL1norm_sma_segLenStddev
20. audspec_lengthL1norm_sma_upleveltime25
21. audspec_lengthL1norm_sma_upleveltime50
22. audspec_lengthL1norm_sma_upleveltime75
23. audspec_lengthL1norm_sma_upleveltime90
24. audspec_lengthL1norm_sma_risetime

ANEXO B: CARACTERÍSTICAS OPENSMILE

25. audspec_lengthL1norm_sma_leftctime
26. audspec_lengthL1norm_sma_lpgain
27. audspec_lengthL1norm_sma_lpc0
28. audspec_lengthL1norm_sma_lpc1
29. audspec_lengthL1norm_sma_lpc2
30. audspec_lengthL1norm_sma_lpc3
31. audspec_lengthL1norm_sma_lpc4

Esto es una manera de estructurar las características extraídas para facilitar su manejo.

	CARACTERÍSTICA	BLOQUE	SUBCARACTERÍSTICAS
1	audspec_lengthL1norm_sma_	1	..range
2	audspecRasta_lengthL1norm_sma_	1	..maxPos
3	pcm_RMSenergy_sma_	1	..minPos
4	pcm_zcr_sma_	1	..quartile1
5	audspec_lengthL1norm_sma_de_	1	..quartile2
6	audspecRasta_lengthL1norm_sma_de_	1	..quartile3
7	pcm_RMSenergy_sma_de_	1	..iqr1-2
8	pcm_zcr_sma_de_	1	..iqr2-3
9	audSpec_Rfilt_sma[0]	1	..iqr1-3
10	audSpec_Rfilt_sma [1]	1	..percentile1.0
11	audSpec_Rfilt_sma [2]	1	..percentile99.0
12	audSpec_Rfilt_sma [3]	1	..pctlrange0-1
13	audSpec_Rfilt_sma [4]	1	..stddev
14	audSpec_Rfilt_sma [5]	1	..skewness
15	audSpec_Rfilt_sma [6]	1	..kurtosis
16	audSpec_Rfilt_sma [7]	1	..meanSegLen
17	audSpec_Rfilt_sma [8]	1	..maxSegLen
18	audSpec_Rfilt_sma [9]	1	..minSegLen
19	audSpec_Rfilt_sma [10]	1	..segLenStddev
20	audSpec_Rfilt_sma [11]	1	..upleveltime25
21	audSpec_Rfilt_sma [12]	1	..upleveltime50
22	audSpec_Rfilt_sma [13]	1	..upleveltime75
23	audSpec_Rfilt_sma [14]	1	..upleveltime90
24	audSpec_Rfilt_sma [15]	1	..risetime
25	audSpec_Rfilt_sma [16]	1	..leftctime
26	audSpec_Rfilt_sma [17]	1	..lpgain
27	audSpec_Rfilt_sma [18]	1	..lpc0
28	audSpec_Rfilt_sma [19]	1	..lpc1
29	audSpec_Rfilt_sma [20]	1	..lpc2
			..lpc3

ANEXO B: CARACTERÍSTICAS OPENSIMILE

30	audSpec_Rfilt_sma [21]	1	..lpc4
31	audSpec_Rfilt_sma [22]	1	
32	audSpec_Rfilt_sma [23]	1	
33	audSpec_Rfilt_sma [24]	1	
34	audSpec_Rfilt_sma [25]	1	
35	pcm_fftMag_fband250-650_sma_	1	
36	pcm_fftMag_fband1000-4000_sma_	1	
37	pcm_fftMag_spectralRollOff25.0_sma_	1	
38	pcm_fftMag_spectralRollOff50.0_sma_	1	
39	pcm_fftMag_spectralRollOff75.0_sma_	1	
40	pcm_fftMag_spectralRollOff90.0_sma_	1	
41	pcm_fftMag_spectralFlux_sma_	1	
42	pcm_fftMag_spectralCentroid_sma_	1	
43	pcm_fftMag_spectralEntropy_sma_	1	
44	pcm_fftMag_spectralVariance_sma_	1	
45	pcm_fftMag_spectralSkewness_sma_	1	
46	pcm_fftMag_spectralKurtosis_sma_	1	
47	pcm_fftMag_spectralSlope_sma_	1	
48	pcm_fftMag_psySharpness_sma_	1	
49	pcm_fftMag_spectralHarmonicity_sma_	1	
50	mfcc_sma[1]	1	
51	mfcc_sma [2]	1	
52	mfcc_sma [3]	1	
53	mfcc_sma [4]	1	
54	mfcc_sma [5]	1	
55	mfcc_sma [6]	1	
56	mfcc_sma [7]	1	
57	mfcc_sma [8]	1	
58	mfcc_sma [9]	1	
59	mfcc_sma [10]	1	
60	mfcc_sma [11]	1	
61	mfcc_sma [12]	1	
62	mfcc_sma [13]	1	
63	mfcc_sma [14]	1	
64	audSpec_Rfilt_sma_de[0]	1	
65	audSpec_Rfilt_sma_de [1]	1	
66	audSpec_Rfilt_sma_de [2]	1	
67	audSpec_Rfilt_sma_de [3]	1	
68	audSpec_Rfilt_sma_de [4]	1	
69	audSpec_Rfilt_sma_de [5]	1	

ANEXO B: CARACTERÍSTICAS OPENSIMILE

70	audSpec_Rfilt_sma_de [6]	1
71	audSpec_Rfilt_sma_de [7]	1
72	audSpec_Rfilt_sma_de [8]	1
73	audSpec_Rfilt_sma_de [9]	1
74	audSpec_Rfilt_sma_de [10]	1
75	audSpec_Rfilt_sma_de [11]	1
76	audSpec_Rfilt_sma_de [12]	1
77	audSpec_Rfilt_sma_de [13]	1
78	audSpec_Rfilt_sma_de [14]	1
79	audSpec_Rfilt_sma_de [15]	1
80	audSpec_Rfilt_sma_de [16]	1
81	audSpec_Rfilt_sma_de [17]	1
82	audSpec_Rfilt_sma_de [18]	1
83	audSpec_Rfilt_sma_de [19]	1
84	audSpec_Rfilt_sma_de [20]	1
85	audSpec_Rfilt_sma_de [21]	1
86	audSpec_Rfilt_sma_de [22]	1
87	audSpec_Rfilt_sma_de [23]	1
88	audSpec_Rfilt_sma_de [24]	1
89	audSpec_Rfilt_sma_de [25]	1
90	pcm_fftMag_fband250-650_sma_de_	1
91	pcm_fftMag_fband1000-4000_sma_de_	1
92	pcm_fftMag_spectralRollOff25.0_sma_de_	1
93	pcm_fftMag_spectralRollOff50.0_sma_de_	1
94	pcm_fftMag_spectralRollOff75.0_sma_de_	1
95	pcm_fftMag_spectralRollOff90.0_sma_de_	1
96	pcm_fftMag_spectralFlux_sma_de_	1
97	pcm_fftMag_spectralCentroid_sma_de_	1
98	pcm_fftMag_spectralEntropy_sma_de_	1
99	pcm_fftMag_spectralVariance_sma_de_	1
100	pcm_fftMag_spectralSkewness_sma_de_	1
101	pcm_fftMag_spectralKurtosis_sma_de_	1
102	pcm_fftMag_spectralSlope_sma_de_	1
103	pcm_fftMag_psySharpness_sma_de_	1
104	pcm_fftMag_spectralHarmonicity_sma_de_	1
105	mfcc_sma_de[1]_	1
106	mfcc_sma_de [2]	1
107	mfcc_sma_de [3]	1
108	mfcc_sma_de [4]	1
109	mfcc_sma_de [5]	1

ANEXO B: CARACTERÍSTICAS OPENSIMILE

110	mfcc_sma_de [6]	1	
111	mfcc_sma_de [7]	1	
112	mfcc_sma_de [8]	1	
113	mfcc_sma_de [9]	1	
114	mfcc_sma_de [10]	1	
115	mfcc_sma_de [11]	1	
116	mfcc_sma_de [12]	1	
117	mfcc_sma_de [13]	1	
118	mfcc_sma_de [14]	1	
1	F0final_sma_	2	..amean
2	voicingFinalUnclipped_sma_	2	..flatness
3	jitterLocal_sma_	2	..posamean
4	jitterDDP_sma_	2	..rqmean
5	shimmerLocal_sma_	2	..linregc1
6	logHNR_sma_	2	..linregc2
7	F0final_sma_de_	2	..linregerrQ
8	voicingFinalUnclipped_sma_de_	2	..qregc1
9	jitterLocal_sma_de_	2	..qregc2
10	jitterDDP_sma_de_	2	..qregc3
11	shimmerLocal_sma_de_	2	..qregerrQ
12	logHNR_sma_de_	2	..centroid ..range ..maxPos ..minPos ..quartile1 ..quartile2 ..quartile3 ..iqr1-2 ..iqr2-3 ..iqr1-3 ..percentile1.0 ..percentile99.0 ..pctlrange0-1 ..stddev ..skewness ..kurtosis ..upleveltime25 ..upleveltime50 ..upleveltime75 ..upleveltime90 ..risetime

ANEXO B: CARACTERÍSTICAS OPENSMILE

			..leftctime ..lpgain ..lpc0 ..lpc1 ..lpc2 ..lpc3 ..lpc4
1	F0final_sma_ff0_	3	..nnz ..meanSegLen ..maxSegLen ..minSegLen ..segLenStddev
1	audspec_lengthL1norm_sma_	4	..amean
2	audspecRasta_lengthL1norm_sma_	4	..flatness
3	pcm_RMSenergy_sma_	4	..rqmean
4	pcm_zcr_sma_	4	..meanPeakDist
5	audSpec_Rfilt_sma[0]	4	..peakDistStddev
6	audSpec_Rfilt_sma [1]	4	..peakRangeAbs
7	audSpec_Rfilt_sma [2]	4	..peakRangeRel
8	audSpec_Rfilt_sma [3]	4	..peakMeanAbs
9	audSpec_Rfilt_sma [4]	4	..peakMeanMeanDist
10	audSpec_Rfilt_sma [5]	4	..peakMeanRel
11	audSpec_Rfilt_sma [6]	4	..minRangeRel
12	audSpec_Rfilt_sma [7]	4	..meanRisingSlope
13	audSpec_Rfilt_sma [8]	4	..stddevRisingSlope
14	audSpec_Rfilt_sma [9]	4	..meanFallingSlope
15	audSpec_Rfilt_sma [10]	4	..stddevFallingSlope
16	audSpec_Rfilt_sma [11]	4	..linregc1
17	audSpec_Rfilt_sma [12]	4	..linregc2
18	audSpec_Rfilt_sma [13]	4	..linregerrQ
19	audSpec_Rfilt_sma [14]	4	..qregc1
20	audSpec_Rfilt_sma [15]	4	..qregc2
21	audSpec_Rfilt_sma [16]	4	..qregc3
22	audSpec_Rfilt_sma [17]	4	..qregerrQ
23	audSpec_Rfilt_sma [18]	4	..centroid
24	audSpec_Rfilt_sma [19]	4	
25	audSpec_Rfilt_sma [20]	4	
26	audSpec_Rfilt_sma [21]	4	
27	audSpec_Rfilt_sma [22]	4	
28	audSpec_Rfilt_sma [23]	4	

ANEXO B: CARACTERÍSTICAS OPENSMILE

29	audSpec_Rfilt_sma [24]	4	
30	audSpec_Rfilt_sma [25]	4	
31	pcm_fftMag_fband250-650_sma_	4	
32	pcm_fftMag_fband1000-4000_sma_	4	
33	pcm_fftMag_spectralRollOff25.0_sma_	4	
34	pcm_fftMag_spectralRollOff50.0_sma_	4	
35	pcm_fftMag_spectralRollOff75.0_sma_	4	
36	pcm_fftMag_spectralRollOff90.0_sma_	4	
37	pcm_fftMag_spectralFlux_sma_	4	
38	pcm_fftMag_spectralCentroid_sma_	4	
39	pcm_fftMag_spectralEntropy_sma_	4	
40	pcm_fftMag_spectralVariance_sma_	4	
41	pcm_fftMag_spectralSkewness_sma_	4	
42	pcm_fftMag_spectralKurtosis_sma_	4	
43	pcm_fftMag_spectralSlope_sma_	4	
44	pcm_fftMag_psySharpness_sma_	4	
45	pcm_fftMag_spectralHarmonicity_sma_	4	
46	mfcc_sma [1]	4	
47	mfcc_sma [2]	4	
48	mfcc_sma [3]	4	
49	mfcc_sma [4]	4	
50	mfcc_sma [5]	4	
51	mfcc_sma [6]	4	
52	mfcc_sma [7]	4	
53	mfcc_sma [8]	4	
54	mfcc_sma [9]	4	
55	mfcc_sma [10]	4	
56	mfcc_sma [11]	4	
57	mfcc_sma [12]	4	
58	mfcc_sma [13]	4	
59	mfcc_sma [14]	4	
1	audspec_lengthL1norm_sma_de_	5	..flatness
2	audspecRasta_lengthL1norm_sma_de_	5	..posamean
3	pcm_RMSenergy_sma_de_	5	..rqmean
4	pcm_zcr_sma_de_	5	..meanPeakDist
5	audSpec_Rfilt_sma_de [0]	5	..peakDistStddev
6	audSpec_Rfilt_sma_de [1]	5	..peakRangeAbs
7	audSpec_Rfilt_sma_de [2]	5	..peakRangeRel
8	audSpec_Rfilt_sma_de [3]	5	..peakMeanAbs
9	audSpec_Rfilt_sma_de [4]	5	..peakMeanMeanDist

ANEXO B: CARACTERÍSTICAS OPENSIMILE

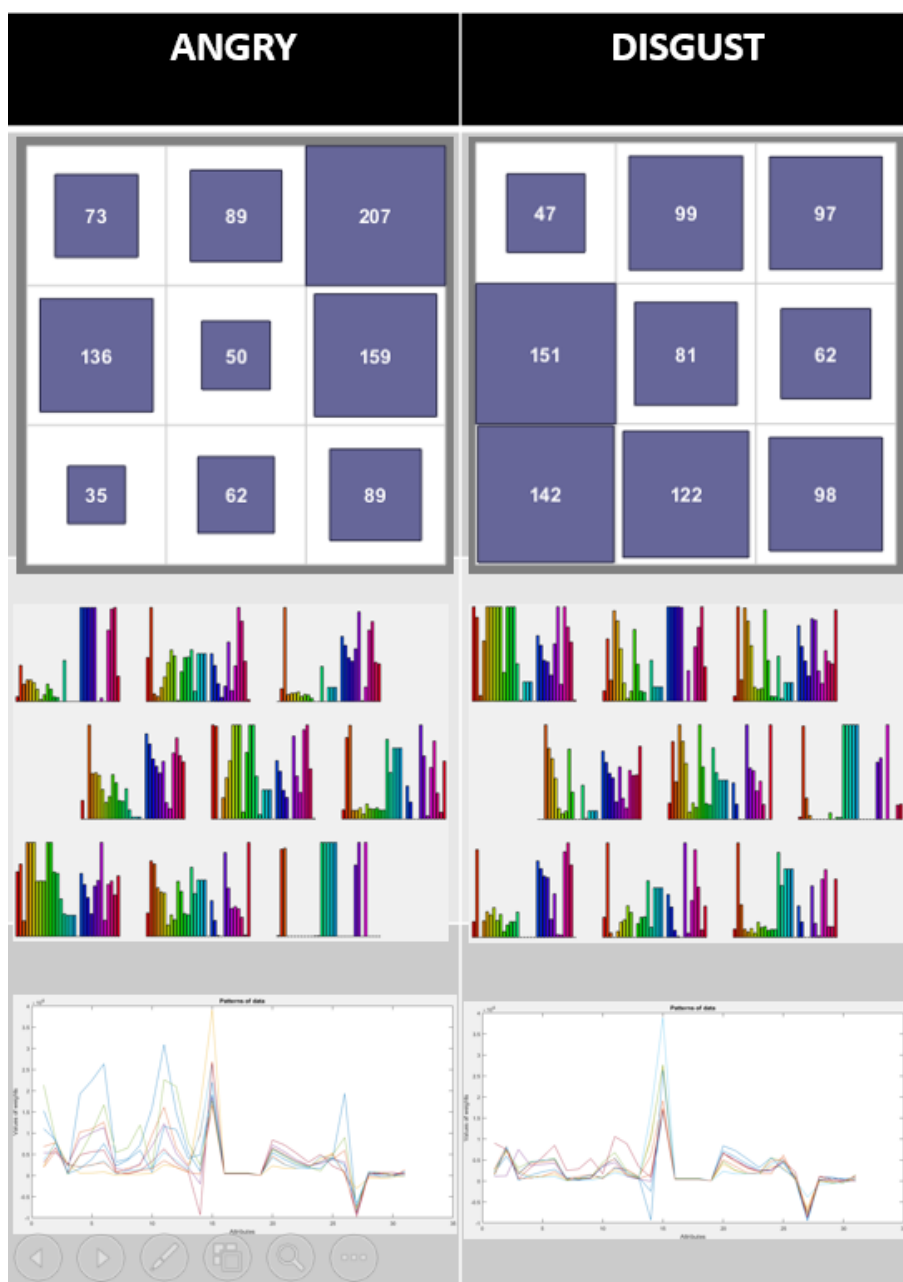
10	audSpec_Rfilt_sma_de [5]	5	..peakMeanRel
11	audSpec_Rfilt_sma_de [6]	5	..minRangeRel
12	audSpec_Rfilt_sma_de [7]	5	..meanRisingSlope
13	audSpec_Rfilt_sma_de [8]	5	..stddevRisingSlope
14	audSpec_Rfilt_sma_de [9]	5	..meanFallingSlope
15	audSpec_Rfilt_sma_de [10]	5	..stddevFallingSlope
16	audSpec_Rfilt_sma_de [11]	5	
17	audSpec_Rfilt_sma_de [12]	5	
18	audSpec_Rfilt_sma_de [13]	5	
19	audSpec_Rfilt_sma_de [14]	5	
20	audSpec_Rfilt_sma_de [15]	5	
21	audSpec_Rfilt_sma_de [16]	5	
22	audSpec_Rfilt_sma_de [17]	5	
23	audSpec_Rfilt_sma_de [18]	5	
24	audSpec_Rfilt_sma_de [19]	5	
25	audSpec_Rfilt_sma_de [20]	5	
26	audSpec_Rfilt_sma_de [21]	5	
27	audSpec_Rfilt_sma_de [22]	5	
28	audSpec_Rfilt_sma_de [23]	5	
29	audSpec_Rfilt_sma_de [24]	5	
30	audSpec_Rfilt_sma_de [25]	5	
31	pcm_fftMag_fband250-650_sma_de_	5	
32	pcm_fftMag_fband1000-4000_sma_de_	5	
33	pcm_fftMag_spectralRollOff25.0_sma_de_	5	
34	pcm_fftMag_spectralRollOff50.0_sma_de_	5	
35	pcm_fftMag_spectralRollOff75.0_sma_de_	5	
36	pcm_fftMag_spectralRollOff90.0_sma_de_	5	
37	pcm_fftMag_spectralFlux_sma_de_	5	
38	pcm_fftMag_spectralCentroid_sma_de_	5	
39	pcm_fftMag_spectralEntropy_sma_de_	5	
40	pcm_fftMag_spectralVariance_sma_de_	5	
41	pcm_fftMag_spectralSkewness_sma_de_	5	
42	pcm_fftMag_spectralKurtosis_sma_de_	5	
43	pcm_fftMag_spectralSlope_sma_de_	5	
44	pcm_fftMag_psySharpness_sma_de_	5	
45	pcm_fftMag_spectralHarmonicity_sma_de_	5	
46	mfcc_sma_de[1]	5	
47	mfcc_sma_de [2]	5	
48	mfcc_sma_de [3]	5	
49	mfcc_sma_de [4]	5	

ANEXO B: CARACTERÍSTICAS OPENSMILE

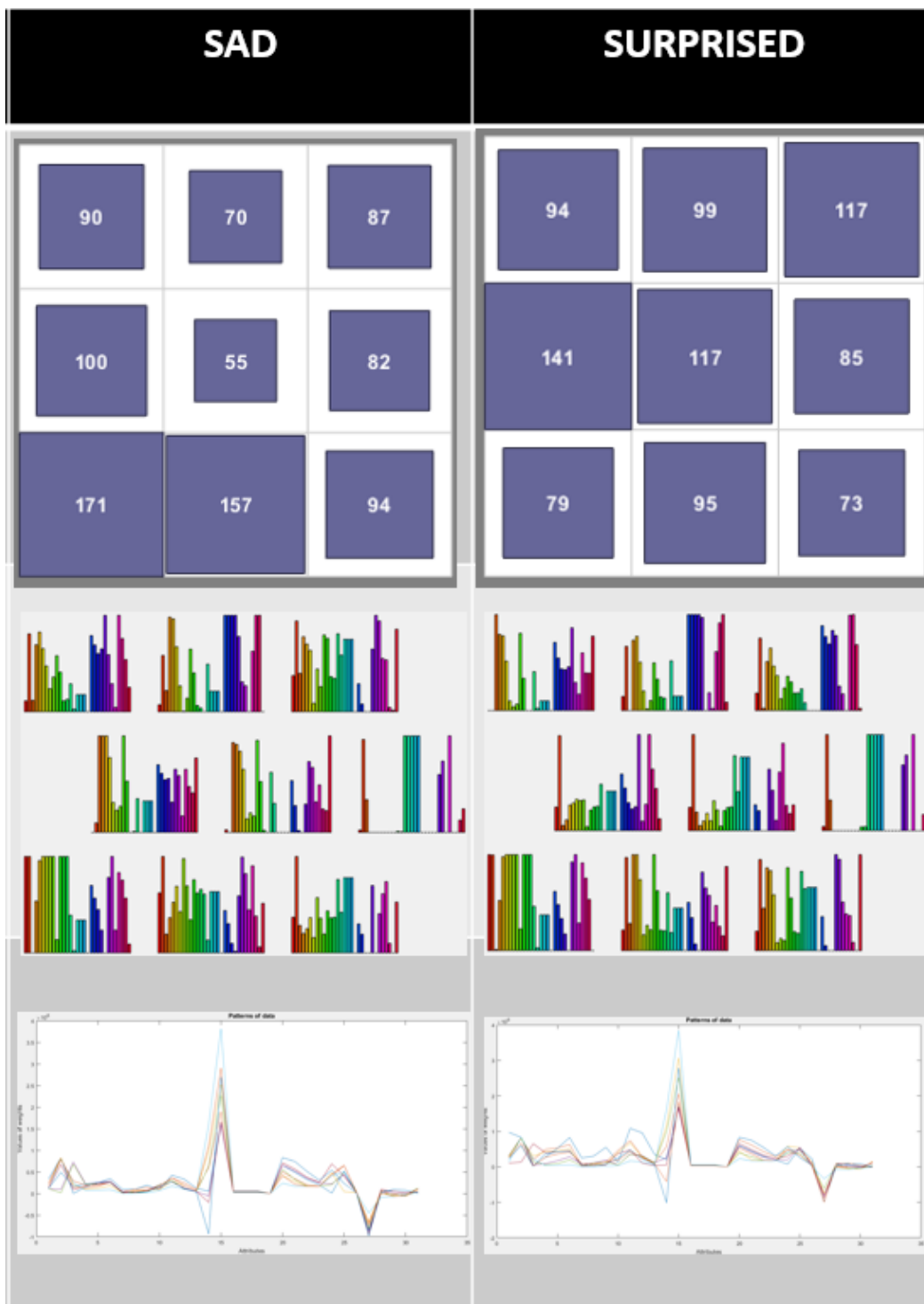
50	mfcc_sma_de [5]	5	
51	mfcc_sma_de [6]	5	
52	mfcc_sma_de [7]	5	
53	mfcc_sma_de [8]	5	
54	mfcc_sma_de [9]	5	
55	mfcc_sma_de [10]	5	
56	mfcc_sma_de [11]	5	
57	mfcc_sma_de [12]	5	
58	mfcc_sma_de [13]	5	
59	mfcc_sma_de [14]	5	

Tabla 7. Características openSMILE organizadas en bloques.

ANEXO C: SOMs CARACTERÍSTICAS BLOQUE1_1



ANEXO C: SOMs CARACTERÍSTICAS BLOQUE1_1



ANEXO C: SOMs CARACTERÍSTICAS BLOQUE1_1



ANEXO D: EJEMPLO ARCHIVO ARFF

```
@RELATION emocion_bloque1_1

@ATTRIBUTE audspec_lengthLlnorm_sma_range REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_maxPos REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_minPos REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_quartile1 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_quartile2 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_quartile3 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_iqr1-2 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_iqr2-3 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_iqr1-3 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_percentile1.0 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_percentile99.0 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_pctlrange0-1 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_stddev REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_skewness REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_kurtosis REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_meanSegLen REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_maxSegLen REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_minSegLen REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_segLenStddev REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_upleveltime25 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_upleveltime50 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_upleveltime75 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_upleveltime90 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_risetime REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_leftctime REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_lpgain REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_lpc0 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_lpc1 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_lpc2 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_lpc3 REAL
@ATTRIBUTE audspec_lengthLlnorm_sma_lpc4 REAL
@ATTRIBUTE class {A, D, SA, SU, N}

@DATA
```

ANEXO D: EJEMPLO ARCHIVO ARFF

1538724.138,769862.069,74689.65517,1963448.276,2292758.621,2674827.586,32
8458.6207,381775.8621,709655.1724,1625000,3130689.655,1506137.931,546379.
3103,95389.65517,2222413.793,40344.82759,40344.82759,40344.82759,0,637827
.5862,465517.2414,321965.5172,201379.3103,310345.3448,439655.2414,2001862
.069,-858896.5517,20949.31034,11395.17241,5776.551724,81289.31034,A
725019.2308,768923.0769,198692.3077,1105173.077,1181134.615,1362596.154,7
6067.30769,181469.2308,257636.5385,1026673.077,1730384.615,704384.6154,24
9542.3077,926980.7692,2635961.538,50000,50000,50000,0,467788.4615,266000,
170192.3077,167000,396154.1923,399038.8269,667500,-769538.4615,-
15431.56904,-37229.03846,-38897.71154,132192.3077,A
190943.3333,811641.0256,256423.0769,48355.12821,51941.66667,73076.92308,3
604.361538,21116.96154,24720.83333,45305.64103,228294.8718,183011.0256,68
907.94872,1653717.949,3930897.436,58974.35897,58974.35897,58974.35897,0,2
20217.9487,167000,167000,167000,410256.6923,538461.7308,16767.13077,-
309923.0769,-69051.15385,-68627.94872,-44257.4359,-8472.602564,A
533343.0681,544683.4739,243689.0748,829611.4622,983629.979,1104631.908,15
4051.3817,121164.9537,275184.3168,667803.8513,1191032.933,523266.0672,189
852.2094,-
201788.0498,1859760.202,35798.31933,35798.31933,35798.31933,0,719865.5462
,550501.4008,371084.0328,243795.5176,267227.3025,369747.958,296626.0664,-
911399.7588,35438.06185,34059.94267,23627.5563,67046.85832,A
2145000,829391.3043,21760.86957,466350,990065.2174,1660347.826,524108.695
7,670086.9565,1194304.348,179254.3478,2286956.522,2108695.652,778304.3478
,282780.4348,1779565.217,43260.86957,43260.86957,43260.86957,0,594304.347
8,434673.913,260913.0435,192260.8696,100000.8696,483695.8696,925608.6957,
-778586.9565,82375.65217,55579.34783,14348.69565,-5832.934783,A
359826.8649,764752.3139,239562.5,224800.4271,254020.9167,360131.3028,2921
5.89313,106075.282,135301.9877,210281.4208,560518.7326,350151.6168,129461
.3697,1007942.276,2679265.708,50694.44444,50694.44444,50694.44444,0,41996
7.5924,250013.8882,166988.4271,166988.4271,445833.6042,399305.9722,82571.
29153,-650020.5431,40170.06993,-34146.35548,-39804.58727,72259.7516,A
293043.4,634213.9299,62164.1791,493369.9627,566514.7925,600523.2687,73182
.59821,33871.91448,107048.2421,343857.5478,628937.5552,285003.8925,100884
.3431,-
942560.2687,2649996.313,34925.37313,34925.37313,34925.37313,0,833014.9239
,719034.8254,529930.3493,315900.497,501492.7612,205223.8806,90803.53209,-
977704.9701,49289.28791,68633.91179,70835.05672,24734.23533,A
1127831.169,833000,36857.14286,161010.3896,399118.1818,772779.2208,238077
.013,373597.4026,611649.3506,66559.74026,1174727.273,1108064.935,413116.8
831,470308.9481,1890519.481,47012.98701,47012.98701,47012.98701,0,539025.
974,385077.9221,233857.1429,177779.2208,150650.1558,399351.013,250274.026
, -725545.4545,90290.5974,42444.79221,8420.701299,-10094.15584,A

ANEXO D: EJEMPLO ARCHIVO ARFF

210976.5602,593328.0423,261924.1624,188480.7804,250249.9206,313830.8323,6
1799.07158,63505.44072,125314.6605,152346.2608,360218.1889,207893.2516,77
928.54503,38804.78619,1665574.603,39259.25926,39259.25926,39259.25926,0,6
45627.8661,487629.6296,327077.6016,228520.2825,314286.0847,502645.5238,36
006.52608,-863826.9878,51347.71783,46167.10774,708.4140741,45125.97741,A
212056.4453,695842.1044,30710.52632,245368.3815,318870.5242,360323.9498,7
3527.05575,41443.25003,114990.9198,175916.76,384211.8877,208205.6535,7696
1.66795,-
414131.6457,1783817.781,33157.89474,33157.89474,33157.89474,0,758739.764,
608385.9667,442918.1281,269040.9368,435088.0877,337719.2982,34761.8285,-
954940.6605,52217.43775,95445.29955,43275.18202,10170.72035,D
210691.3104,819666.6648,71030.8642,123125.7057,176712.6409,250735.9079,53
615.26884,74045.10507,127679.6452,100488.3053,308196.2821,207740.8777,785
92.72,302348.8124,1672420.346,43827.16049,43827.16049,43827.16049,0,58654
5.2685,410323.0444,263323.0451,194660.4944,204938.9691,592592.6543,23682.
5456,-818521.437,78322.60596,29799.44667,-20690.81889,49326.19066,D
179739.7496,760233.0078,160187.7019,67379.7623,81887.18547,140225.411,145
01.85158,58392.77805,72911.95996,61990.81016,237259.1373,175298.5209,6564
4.20151,1002885.929,2584162.117,51844.66019,51844.66019,51844.66019,0,402
728.1544,262084.1427,166970.8767,166970.8767,244660.7573,373786.8738,1416
9.46563,-637415.8913,64246.85309,-44921.60138,-30923.9488,52048.84845,D
429797.4333,349238.0905,499825.3952,893595.1524,1034548.19,1121160.738,14
0010.181,87179.2581,227341.819,741811.219,1165323.286,422226.3,155317.447
6,-
554647.2619,1971337.048,38095.2381,38095.2381,38095.2381,0,785571.4238,64
3095.2429,484142.8571,309476.1857,247619.2381,273809.5238,292599.5619,-
933449.2,36131.91238,43365.34286,25535.96814,71283.50238,D
98571.13711,158450.7042,726455.3979,347452.343,375484.1989,403502.7203,28
021.22007,28017.46546,56024.67162,327735.5634,424612.8641,96980.8219,3600
1.64708,-
435.4175352,1691837.028,38521.12676,38521.12676,38521.12676,0,669075.1183
,497650.2345,334460.0937,231330.988,253521.1268,468309.9085,51591.33718,-
879473.5169,13819.84303,6449.568465,-1167.162,109677.2558,D
130927.2351,796269.6059,210828.4309,120725.4375,125448.8691,141055.9121,4
693.709762,15646.42569,20330.17943,115181.8145,240874.1963,125655.764,460
26.83074,1564274.897,3776893.794,57647.05882,57647.05882,57647.05882,0,26
2333.3324,166975.4926,166975.4926,166975.4926,405882.6912,496323.7941,232
90.83435,-452823.0809,-65003.35781,-62434.12221,-
38163.98426,39626.21338,D
804777.9945,814896.9691,24284.84909,209242.5382,412309.5145,680993.6655,2
03039.7036,268738.6964,471891.1273,110971.1655,903888.7109,792946.6364,29
7026.1909,258896.1764,1701448.145,43454.54545,43454.54545,43454.54545,0,5

ANEXO D: EJEMPLO ARCHIVO ARFF

91036.3636,433206.06,269642.4236,200212.1218,218182.5455,527272.8182,1555
46.6249,-807504.4818,106796.2236,45330.7792,2017.720364,4575.392909,D
168511.6905,603732.2393,136562.841,393979.6954,431295.6515,447901.5062,37
345.95413,16551.10167,53874.92475,304602.2597,468005.6328,163439.4377,570
03.77721,-
995201.3787,2779005.311,39836.06557,39836.06557,39836.06557,0,833038.2475
,737743.1689,524677.5951,325174.8623,573770.5738,278688.5246,54419.62384,
-954396.0098,37866.00339,44811.58443,51664.65032,58584.30316,D
222158.0148,721969.346,279624.5207,437773.7069,463840.4046,520940.0782,25
962.82138,57114.96299,83057.40552,418617.6943,634840.4782,216118.1851,770
94.35448,875221.0161,2562857.034,44252.87356,44252.87356,44252.87356,0,49
2279.692,289176.2448,168881.2287,166973.1828,535632.2874,454023.3218,1013
25.5055,-787033.2644,-2459.610483,-34142.84517,-48091.53989,140458.4976,D
320098.1737,833008.3325,4166.666875,141769.0142,230432.5322,323165.8721,8
8645.39313,92660.83919,181318.1073,83069.38269,398124.2948,315044.5998,11
5272.923,61771.48375,1650888.906,40187.5,40187.5,40187.5,0,652245.8338,49
0608.3331,313399.9994,199164.5838,1250.99375,517187.5438,36830.41512,-
842612.0256,59570.85519,47464.0995,25509.62362,16317.51813,SA
247067.1749,829258.1444,135413.5338,124515.2077,165831.7987,259760.9353,4
1330.68866,93939.63961,135255.8772,111902.2777,354706.8337,242722.3756,92
051.59857,607496.8459,1907928.639,48421.05263,48421.05263,48421.05263,0,5
03749.3737,340543.8586,201937.3444,168238.0962,416541.7669,667293.2932,34
748.52079,-742867.7278,80941.13925,-10277.47097,-
52145.0067,70162.20947,SA
190249.8786,823091.269,204325.3976,97187.675,116221.5685,176446.9845,1901
3.26274,60237.91571,79244.66631,89497.92774,275166.9202,185643.8738,68713
.5769,980137.5214,2538530.321,52619.04762,52619.04762,52619.04762,0,41450
3.9679,273722.2214,166992.0643,166992.0643,454762.2024,47619.94048,21138.
91587,-653917.6167,42135.0856,-50896.58327,-61370.53488,70200.72238,SA
100843.2268,117256.411,730721.611,218463.689,249434.8923,281203.0714,3101
9.16923,31607.86058,62643.33769,199453.2615,298963.6209,99449.81011,37832
.85385,-
63175.3002,1606936.308,38681.31868,38681.31868,38681.31868,0,661271.0637,
518344.3231,358824.1747,245483.5165,246153.8462,442307.7363,26925.94179,-
884401.8604,27651.64901,8501.736352,-3232.808791,101208.5087,SA
100008.6281,13916.66667,673583.3333,195377.5708,211546.0738,236772.5165,1
6121.75263,25296.76663,41431.42313,182558.7506,280077.0033,97546.79438,34
131.69417,616481.0125,2256862.229,31041.66667,31041.66667,31041.66667,0,5
72965.2771,319243.0542,177361.1125,166986.1125,200000,505208.5833,24892.6
4413,-809255.2646,88.948125,-28963.48852,-27366.36729,127230.7285,SA
121166.576,797309.5229,250019.0471,86616.88571,91508.01057,105506.1997,48
96.881086,13986.06014,18883.16986,82246.20614,198619.16,116305.0976,42651
.02186,1586725.186,3806282.614,59142.85714,59142.85714,59142.85714,0,2476

ANEXO D: EJEMPLO ARCHIVO ARFF

61.9043,166990.4771,166990.4771,166990.4771,420000.2857,482143.1143,12833
.39254,-456411.7443,-76686.77436,-67164.74729,-41900.13757,38247.80857,SA
136512.6437,668799.1449,76995.72692,183397.5168,231659.1004,266594.7586,4
8221.57013,35033.73564,83265.83141,151663.8676,286228.1569,134598.7753,51
210.11782,-
223527.2454,1667602.5,35512.82051,35512.82051,35512.82051,0,711576.9244,5
70645.3,382363.2474,271384.6141,697435.8974,375000.0256,20009.82385,-
958190.1038,92094.19256,62722.34577,19372.00526,31115.05872,SA
123966.8225,479426.9,184157.8947,232527.2893,260939.1653,274991.3381,2844
8.01737,14069.52298,42540.54035,166498.7118,286846.8007,120325.2819,42401
.32526,-
957619.7246,2741880.474,37543.85965,37543.85965,37543.85965,0,833017.5421
,748520.4684,508853.8018,321666.6649,540350.8947,228070.1754,19743.66762,
-976504.9,30494.98421,86337.81965,75131.19386,14415.65088,SA
189642.6824,783037.4988,314504.1663,198385.8655,211029.1308,258013.066,12
658.47773,47008.75525,59656.33788,183867.5426,367829.1029,183981.3115,659
53.21038,1115526.263,2938125.525,51000,51000,51000,0,422762.4988,206412.5
013,166987.5013,166987.5013,475000.175,659375.0375,39553.1507,-
692327.5425,3902.357,-62463.61263,-58643.43875,115859.8033,SA
992712.1212,830484.8485,25257.57576,290159.0909,546227.2727,854636.3636,2
56181.8182,308075.7576,564166.6667,139743.9394,1116121.212,976545.4545,36
0727.2727,181480.303,1712878.788,42575.75758,42575.75758,42575.75758,0,61
1242.4242,456984.8485,300424.2424,194666.6667,87879.66667,503787.9545,235
510.6061,-812045.4545,82518.0303,59373.93939,14836.06061,-4782.166667,SU
290851.1111,812633.3333,181544.4444,489011.1111,532300,626500,43328.88889
,94304.11111,137598.7778,458077.7778,742677.7778,284655.5556,103483,58830
0,2068555.556,45222.22222,45222.22222,45222.22222,0,533322.2222,364544.44
44,194666.6667,167000,491111.3889,469444.7222,130804.4444,-
801122.2222,7701.877778,-28607.83333,-41347,131061.1111,SU
279520.9677,795387.0968,298290.3226,416693.5484,440693.5484,494967.7419,2
4037.74194,54250.48387,78261.77419,389827.4194,660338.7097,270332.2581,94
424.51613,1151693.548,3068387.097,49516.12903,49516.12903,49516.12903,0,4
54193.5484,199129.0323,167000,167000,580645.1935,524193.7581,112301.7742,
-741467.7419,-9147.130645,-58781.29032,-66252.35484,148109.6774,SU
268208.1081,680099.0991,33045.04505,253518.9189,348510.8108,403330.6306,9
4950.09009,54830.40541,149801.6216,169485.5856,433280.1802,263772.0721,97
966.12613,-
409879.2793,1804054.054,33063.06306,33063.06306,33063.06306,0,759720.7207
,614342.3423,430864.8649,282234.2342,448648.991,335585.5856,43533.10811,-
991684.6847,90500.81081,105743.7477,38240.96396,-4498.108108,SU
301587.0192,808971.1538,142615.3846,59025.28846,87303.84615,188536.5385,2
8262.55769,101201.5865,129437,52176.44231,346560.5769,294377.7885,110556.
5096,988326.9231,2524326.923,52788.46154,52788.46154,52788.46154,0,397230

ANEXO D: EJEMPLO ARCHIVO ARFF

.7692,285115.3846,167000,167000,388461.9615,355769.6827,29679.30865,-
605307.6923,73191.53846,-27592.01923,-40967.88462,37064.35577,SU
137784.9315,821589.0411,223726.0274,37830,40892.32877,57326.0274,3057.041
096,16428.31507,19490.24658,34938.08219,167045.2055,132142.4658,49450.410
96,1627260.274,3881095.89,59315.06849,59315.06849,59315.06849,0,223890.41
1,167000,167000,167000,400000.3288,500000.2192,8151.508219,-330426.0274,-
76533.69863,-74376.16438,-40386,-3313.027397,SU
106058.2558,126011.6279,658883.7209,401920.9302,429658.1395,460038.3721,2
7768.37209,30298.62791,58090.23256,380105.814,484367.4419,104206.6279,381
29.06977,55637.90698,1701860.465,35697.67442,35697.67442,35697.67442,0,65
7046.5116,482523.2558,325476.7442,221046.5116,306976.7442,488372.1163,727
11.36279,-884244.186,26794.40698,-148.6430233,-5561.232558,112050,SU
233844.1489,595638.2979,14180.85106,384029.7872,437951.0638,462521.2766,5
4004.04255,24466.8617,78453.40426,260414.8936,487534.0426,227070.4255,794
27.44681,-
990031.9149,2753617.021,37021.2766,37021.2766,37021.2766,0,833000,746468.
0851,533755.3191,296117.0213,568085.2128,260638.2979,53294.23723,-
984861.7021,35752.76596,89506.67447,77618.75532,11097.34043,SU
318606.6387,827420.1681,40680.67227,89127.89916,172117.1429,280919.3277,8
3009.94958,108846.8908,191832.1849,53779.41176,367790.7563,313981.4286,11
8225.4622,266116.8151,1645798.319,44369.7479,44369.7479,44369.7479,0,5813
86.5546,424218.4874,274420.1681,194899.1597,169748.6723,565126.084,30187.
56975,-810201.6807,93498.47899,40222.26891,-10.42016807,13791.09244,SU
363992.5926,833000,0,117209.1852,214484.4444,324133.3333,97254.81481,1096
95.5556,207009.6296,58403.25926,416533.3333,358162.2222,131082.2222,14974
3.1111,1644296.296,42000,42000,42000,0,631066.6667,456703.7037,292429.629
6,192822.2222,1,537037.0519,36312.26667,-
822429.6296,74046.71111,49516.96296,22052.51852,2608.148148,N
206937.931,833000,0,33140.68966,61708.04598,134378.1609,28568.22989,72727
.47126,101273.2184,25016.2069,227704.5977,202644.8276,76243.44828,779517.
2414,2163793.103,51379.31034,51379.31034,51379.31034,0,448160.9195,327298
.8506,178448.2759,167000,1,540230.069,11586.22529,-
657252.8736,94647.06897,-11329.65517,-30171.49425,25221.63218,N
168641.9728,782006.8027,340034.0136,147214.1497,163006.1905,208132.6531,1
5800.17755,45112.58503,60927.41497,136892.3129,300902.0408,163974.2177,59
300.68027,1042578.231,2753129.252,51972.78912,51972.78912,51972.78912,0,4
17054.4218,239272.1088,167000,167000,575510.2109,340136.5442,23350.34218,
-694244.898,25584.55102,-66302.31293,-76655.34014,115968.9116,N
59747.4359,209384.6154,777615.3846,202138.4615,221500,233220.5128,19257.1
7949,11750.41026,31002.30769,182769.2308,241364.1026,58653.07692,21325.89
744,-
451179.4872,1837435.897,42051.28205,42051.28205,42051.28205,0,764897.4359

ANEXO D: EJEMPLO ARCHIVO ARFF

```
, 628461.5385, 427282.0513, 256435.8974, 266666.6667, 423076.9231, 16982.05128,
-897384.6154, -11562.58974, 38650.25641, 38779.22821, 73372.05128, N
211563.0189, 826716.9811, 243886.7925, 134441.0377, 172102.8302, 252667.9245, 3
7708.58491, 80533.01887, 118302.8302, 122109.5283, 330192.4528, 208052.8302, 78
914.15094, 567565, 1886698.113, 47358.49057, 47358.49057, 47358.49057, 0, 506273
.5849, 355056.6038, 210849.0566, 171698.1132, 698113.2075, 639151.0283, 25196.6
8962, -759584.9057, 59514.62264, -12113.9717, -61072.34906, 84777.35849, N
98428.46154, 811641.0256, 252141.0256, 68830.89744, 72947.94872, 84010.64103, 4
135.358974, 11082.33333, 15220.21795, 65344.74359, 159817.9487, 94464.74359, 34
688.84615, 1589871.795, 3829358.974, 58205.12821, 58205.12821, 58205.12821, 0, 2
52205.1282, 167000, 167000, 167000, 379487.5128, 580128.3333, 8004.038462, -
443910.2564, -78729.62821, -68057.82051, -42875.83333, 39145.25641, N
156437.7922, 647155.8442, 15155.84416, 119626.4935, 175916.8831, 207720.7792, 5
6231.81818, 31882.07792, 88097.14286, 71831.81818, 225724.6753, 153946.4935, 57
588.31169, -
429153.2468, 1797532.468, 32987.01299, 32987.01299, 32987.01299, 0, 766155.8442
, 614948.0519, 432844.1558, 290000, 527272.9351, 324675.3247, 10308.40519, -
1017103.896, 125072.8571, 115051.1688, 40478.46753, -23794.80519, N
111638.4615, 499861.5385, 58969.23077, 162247.6923, 184513.8462, 195943.0769, 2
2253.53846, 11451.07692, 33714.46154, 102067.6923, 210196.9231, 108135.3846, 37
264.30769, -
1010569.231, 2857384.615, 38000, 38000, 38000, 0, 833000, 758923.0769, 525723.076
9, 300046.1538, 538461.5692, 257692.3077, 8508.769231, -
1007861.538, 35507.23077, 116291.8154, 98923.23077, -21555.53846, N
64750, 136231.7073, 691085.3659, 168153.6585, 185076.8293, 205264.6341, 16888.4
1463, 20187.14634, 37074.87805, 156836.5854, 220547.561, 63679.26829, 23585.121
95, 155621.9512, 1617560.976, 33658.53659, 33658.53659, 33658.53659, 0, 632256.0
976, 459268.2927, 312902.439, 223707.3171, 280487.8049, 551829.3049, 12646.1585
4, -858926.8293, 11004.62195, -1911.060976, -13672.5, 116101.2195, N
```

ANEXO E: TABLA FIABILIDAD Y ERROR

BLOQUE 1

	CARACTERÍSTICA	CORRECTO	ERROR
1	audspec_lengthL1norm_sma_	84.4444 %	15.5556 %
2	audspecRasta_lengthL1norm_sma_	86.6667 %	13.3333 %
3	pcm_RMSenergy_sma_	95.5556 %	4.4444 %
4	pcm_zcr_sma_	77.7778 %	22.2222 %
5	audspec_lengthL1norm_sma_de_	91.1111 %	8.8889 %
6	audspecRasta_lengthL1norm_sma_de_	84.4444 %	15.5556 %
7	pcm_RMSenergy_sma_de_	88.8889 %	11.1111 %
8	pcm_zcr_sma_de_	82.2222 %	17.7778 %
9	audSpec_Rfilt_sma[0]	75.5556 %	24.4444 %
10	audSpec_Rfilt_sma [1]	86.6667 %	13.3333 %
11	audSpec_Rfilt_sma [2]	86.6667 %	13.3333 %
12	audSpec_Rfilt_sma [3]	93.3333 %	6.6667 %
13	audSpec_Rfilt_sma [4]	80 %	20 %
14	audSpec_Rfilt_sma [5]	88.8889 %	11.1111 %
15	audSpec_Rfilt_sma [6]	86.6667 %	13.3333 %
16	audSpec_Rfilt_sma [7]	86.6667 %	13.3333 %
17	audSpec_Rfilt_sma [8]	84.4444 %	15.5556 %
18	audSpec_Rfilt_sma [9]	84.4444 %	15.5556 %
19	audSpec_Rfilt_sma [10]	84.4444 %	15.5556 %
20	audSpec_Rfilt_sma [11]	95.5556 %	4.4444 %
21	audSpec_Rfilt_sma [12]	84.4444 %	15.5556 %
22	audSpec_Rfilt_sma [13]	84.4444 %	15.5556 %
23	audSpec_Rfilt_sma [14]	86.6667 %	13.3333 %
24	audSpec_Rfilt_sma [15]	86.6667 %	13.3333 %
25	audSpec_Rfilt_sma [16]	95.5556 %	4.4444 %
26	audSpec_Rfilt_sma [17]	84.4444 %	15.5556 %
27	audSpec_Rfilt_sma [18]	88.8889 %	11.1111 %
28	audSpec_Rfilt_sma [19]	91.1111 %	8.8889 %
29	audSpec_Rfilt_sma [20]	88.8889 %	11.1111 %
30	audSpec_Rfilt_sma [21]	84.4444 %	15.5556 %
31	audSpec_Rfilt_sma [22]	86.6667 %	13.3333 %

ANEXO E: TABLA FIABILIDAD Y ERROR

32	audSpec_Rfilt_sma [23]	84.4444 %	15.5556 %
33	audSpec_Rfilt_sma [24]	86.6667 %	13.3333 %
34	audSpec_Rfilt_sma [25]	82.2222 %	17.7778 %
35	pcm_fftMag_fband250-650_sma_	86.6667 %	13.3333 %
36	pcm_fftMag_fband1000-4000_sma_	77.7778 %	22.2222 %
37	pcm_fftMag_spectralRollOff25.0_sma_	80 %	20 %
38	pcm_fftMag_spectralRollOff50.0_sma_	88.8889 %	11.1111 %
39	pcm_fftMag_spectralRollOff75.0_sma_	91.1111 %	8.8889 %
40	pcm_fftMag_spectralRollOff90.0_sma_	88.8889 %	11.1111 %
41	pcm_fftMag_spectralFlux_sma_	93.3333 %	6.6667 %
42	pcm_fftMag_spectralCentroid_sma_	86.6667 %	13.3333 %
43	pcm_fftMag_spectralEntropy_sma_	82.2222 %	17.7778 %
44	pcm_fftMag_spectralVariance_sma_	93.3333 %	6.6667 %
45	pcm_fftMag_spectralSkewness_sma_	82.2222 %	17.7778 %
46	pcm_fftMag_spectralKurtosis_sma_	75.5556 %	24.4444 %
47	pcm_fftMag_spectralSlope_sma_	91.1111 %	8.8889 %
48	pcm_fftMag_psySharpness_sma_	86.6667 %	13.3333 %
49	pcm_fftMag_spectralHarmonicity_sma_	93.3333 %	6.6667 %
50	mfcc_sma [1]	88.8889 %	11.1111 %
51	mfcc_sma [2]	82.2222 %	17.7778 %
52	mfcc_sma [3]	84.4444 %	15.5556 %
53	mfcc_sma [4]	80 %	20 %
54	mfcc_sma [5]	84.4444 %	15.5556 %
55	mfcc_sma [6]	82.2222 %	17.7778 %
56	mfcc_sma [7]	88.8889 %	11.1111 %
57	mfcc_sma [8]	80 %	20 %
58	mfcc_sma [9]	91.1111 %	8.8889 %
59	mfcc_sma [10]	86.6667 %	13.3333 %
60	mfcc_sma [11]	82.2222 %	17.7778 %
61	mfcc_sma [12]	86.6667 %	13.3333 %
62	mfcc_sma [13]	84.4444 %	15.5556 %
63	mfcc_sma [14]	84.4444 %	15.5556 %
64	audSpec_Rfilt_sma_de[0]	88.8889 %	11.1111 %
65	audSpec_Rfilt_sma_de [1]	86.6667 %	13.3333 %
66	audSpec_Rfilt_sma_de [2]	84.4444 %	15.5556 %
67	audSpec_Rfilt_sma_de [3]	88.8889 %	11.1111 %
68	audSpec_Rfilt_sma_de [4]	84.4444 %	15.5556 %
69	audSpec_Rfilt_sma_de [5]	84.4444 %	15.5556 %
70	audSpec_Rfilt_sma_de [6]	84.4444 %	15.5556 %
71	audSpec_Rfilt_sma_de [7]	66.6667 %	33.3333 %

ANEXO E: TABLA FIABILIDAD Y ERROR

72	audSpec_Rfilt_sma_de [8]	82.2222 %	17.7778 %
73	audSpec_Rfilt_sma_de [9]	84.4444 %	15.5556 %
74	audSpec_Rfilt_sma_de [10]	82.2222 %	82.2222 %
75	audSpec_Rfilt_sma_de [11]	86.6667 %	13.3333 %
76	audSpec_Rfilt_sma_de [12]	84.4444 %	84.4444 %
77	audSpec_Rfilt_sma_de [13]	86.6667 %	86.6667 %
78	audSpec_Rfilt_sma_de [14]	82.2222 %	17.7778 %
79	audSpec_Rfilt_sma_de [15]	86.6667 %	13.3333 %
80	audSpec_Rfilt_sma_de [16]	88.8889 %	11.1111 %
81	audSpec_Rfilt_sma_de [17]	86.6667 %	13.3333 %
82	audSpec_Rfilt_sma_de [18]	86.6667 %	13.3333 %
83	audSpec_Rfilt_sma_de [19]	82.2222 %	17.7778 %
84	audSpec_Rfilt_sma_de [20]	86.6667 %	13.3333 %
85	audSpec_Rfilt_sma_de [21]	82.2222 %	17.7778 %
86	audSpec_Rfilt_sma_de [22]	88.8889 %	11.1111 %
87	audSpec_Rfilt_sma_de [23]	91.1111 %	8.8889 %
88	audSpec_Rfilt_sma_de [24]	86.6667 %	13.3333 %
89	audSpec_Rfilt_sma_de [25]	86.6667 %	13.3333 %
90	pcm_fftMag_fband250-650_sma_de_	88.8889 %	11.1111 %
91	pcm_fftMag_fband1000-4000_sma_de_	93.3333 %	6.6667 %
92	pcm_fftMag_spectralRollOff25.0_sma_de_	80 %	20 %
93	pcm_fftMag_spectralRollOff50.0_sma_de_	80 %	20 %
94	pcm_fftMag_spectralRollOff75.0_sma_de_	97.7778 %	2.2222 %
95	pcm_fftMag_spectralRollOff90.0_sma_de_	77.7778 %	22.2222 %
96	pcm_fftMag_spectralFlux_sma_de_	88.8889 %	11.1111 %
97	pcm_fftMag_spectralCentroid_sma_de_	91.1111 %	8.8889 %
98	pcm_fftMag_spectralEntropy_sma_de_	86.6667 %	13.3333 %
99	pcm_fftMag_spectralVariance_sma_de_	88.8889 %	11.1111 %
100	pcm_fftMag_spectralSkewness_sma_de_	86.6667 %	13.3333 %
101	pcm_fftMag_spectralKurtosis_sma_de_	84.4444 %	15.5556 %
102	pcm_fftMag_spectralSlope_sma_de_	88.8889 %	11.1111 %
103	pcm_fftMag_psySharpness_sma_de_	82.2222 %	17.7778 %
104	pcm_fftMag_spectralHarmonicity_sma_de_	91.1111 %	8.8889 %
105	mfcc_sma_de [1]	84.4444 %	15.5556 %
106	mfcc_sma_de [2]	82.2222 %	17.7778 %
107	mfcc_sma_de [3]	86.6667 %	13.3333 %
108	mfcc_sma_de [4]	82.2222 %	17.7778 %
109	mfcc_sma_de [5]	84.4444 %	15.5556 %
110	mfcc_sma_de [6]	77.7778 %	22.2222 %
111	mfcc_sma_de [7]	82.2222 %	17.7778 %

ANEXO E: TABLA FIABILIDAD Y ERROR

112	mfcc_sma_de [8]	80 %	20 %
113	mfcc_sma_de [9]	84.4444 %	15.5556 %
114	mfcc_sma_de [10]	88.8889 %	11.1111 %
115	mfcc_sma_de [11]	82.2222 %	17.7778 %
116	mfcc_sma_de [12]	84.4444 %	15.5556 %
117	mfcc_sma_de [13]	91.1111 %	8.8889 %
118	mfcc_sma_de [14]	82.2222 %	17.7778 %

Tabla 8. Tabla fiabilidad/error bloque 1

BLOQUE 2

	CARACTERÍSTICA	CORRECTO	ERROR
1	F0final_sma_	88.8889 %	11.1111 %
2	voicingFinalUnclipped_sma_	80 %	20 %
3	jitterLocal_sma_	75.5556 %	24.4444 %
4	jitterDDP_sma_	77.7778 %	22.2222 %
5	shimmerLocal_sma_	77.7778 %	22.2222 %
6	logHNR_sma_	88.8889 %	11.1111 %
7	F0final_sma_de_	88.8889 %	11.1111 %
8	voicingFinalUnclipped_sma_de_	84.4444 %	15.5556 %
9	jitterLocal_sma_de_	86.6667 %	13.3333 %
10	jitterDDP_sma_de_	80 %	20 %
11	shimmerLocal_sma_de_	68.8889 %	31.1111 %
12	logHNR_sma_de_	82.2222 %	17.7778 %

Tabla 9. Tabla fiabilidad/error bloque 2

BLOQUE 3

	CARACTERÍSTICA	CORRECTO	ERROR
1	F0final_sma_ff0_	20 %	80 %

Tabla 10. Tabla fiabilidad/error bloque 3

BLOQUE 4

	CARACTERÍSTICA	CORRECTO	ERROR
1	audspec_lengthL1norm_sma_	84.4444 %	15.5556 %
2	audspecRasta_lengthL1norm_sma_	82.2222 %	17.7778 %
3	pcm_RMSenergy_sma_	93.3333 %	6.6667 %
4	pcm_zcr_sma_	75.5556 %	24.4444 %
5	audSpec_Rfilt_sma[0]_	80 %	20 %
6	audSpec_Rfilt_sma [1]	62.2222 %	37.7778 %
7	audSpec_Rfilt_sma [2]	73.3333 %	26.6667 %
8	audSpec_Rfilt_sma [3]	86.6667 %	13.3333 %
9	audSpec_Rfilt_sma [4]	77.7778 %	22.2222 %
10	audSpec_Rfilt_sma [5]	75.5556 %	24.4444 %
11	audSpec_Rfilt_sma [6]	75.5556 %	24.4444 %
12	audSpec_Rfilt_sma [7]	73.3333 %	26.6667 %
13	audSpec_Rfilt_sma [8]	75.5556 %	24.4444 %
14	audSpec_Rfilt_sma [9]	64.4444 %	35.5556 %
15	audSpec_Rfilt_sma [10]	68.8889 %	31.1111 %
16	audSpec_Rfilt_sma [11]	77.7778 %	22.2222 %
17	audSpec_Rfilt_sma [12]	62.2222 %	37.7778 %
18	audSpec_Rfilt_sma [13]	84.4444 %	15.5556 %
19	audSpec_Rfilt_sma [14]	82.2222 %	17.7778 %
20	audSpec_Rfilt_sma [15]	82.2222 %	17.7778 %
21	audSpec_Rfilt_sma [16]	77.7778 %	22.2222 %
22	audSpec_Rfilt_sma [17]	80 %	20 %
23	audSpec_Rfilt_sma [18]	71.1111 %	28.8889 %
24	audSpec_Rfilt_sma [19]	35.5556 %	64.4444 %
25	audSpec_Rfilt_sma [20]	75.5556 %	24.4444 %
26	audSpec_Rfilt_sma [21]	68.8889 %	31.1111 %
27	audSpec_Rfilt_sma [22]	73.3333 %	26.6667 %
28	audSpec_Rfilt_sma [23]	80 %	20 %
29	audSpec_Rfilt_sma [24]	73.3333 %	26.6667 %
30	audSpec_Rfilt_sma [25]	75.5556 %	24.4444 %
31	pcm_fftMag_fband250-650_sma_	77.7778 %	22.2222 %
32	pcm_fftMag_fband1000-4000_sma_	82.2222 %	17.7778 %
33	pcm_fftMag_spectralRollOff25.0_sma_	84.4444 %	15.5556 %
34	pcm_fftMag_spectralRollOff50.0_sma_	88.8889 %	11.1111 %
35	pcm_fftMag_spectralRollOff75.0_sma_	86.6667 %	13.3333 %
36	pcm_fftMag_spectralRollOff90.0_sma_	86.6667 %	13.3333 %

ANEXO E: TABLA FIABILIDAD Y ERROR

37	pcm_fftMag_spectralFlux_sma_	91.1111 %	8.8889 %
38	pcm_fftMag_spectralCentroid_sma_	88.8889 %	11.1111 %
39	pcm_fftMag_spectralEntropy_sma_	68.8889 %	31.1111 %
40	pcm_fftMag_spectralVariance_sma_	88.8889 %	11.1111 %
41	pcm_fftMag_spectralSkewness_sma_	75.5556 %	24.4444 %
42	pcm_fftMag_spectralKurtosis_sma_	84.4444 %	15.5556 %
43	pcm_fftMag_spectralSlope_sma_	86.6667 %	13.3333 %
44	pcm_fftMag_psySharpness_sma_	77.7778 %	22.2222 %
45	pcm_fftMag_spectralHarmonicity_sma_	82.2222 %	17.7778 %
46	mfcc_sma[1]_	82.2222 %	17.7778 %
47	mfcc_sma [2]	80 %	20 %
48	mfcc_sma [3]	80 %	20 %
49	mfcc_sma [4]	84.4444 %	15.5556 %
50	mfcc_sma [5]	82.2222 %	17.7778 %
51	mfcc_sma [6]	91.1111 %	8.8889 %
52	mfcc_sma [7]	77.7778 %	22.2222 %
53	mfcc_sma [8]	84.4444 %	15.5556 %
54	mfcc_sma [9]	77.7778 %	22.2222 %
55	mfcc_sma [10]	84.4444 %	15.5556 %
56	mfcc_sma [11]	82.2222 %	17.7778 %
57	mfcc_sma [12]	82.2222 %	17.7778 %
58	mfcc_sma [13]	82.2222 %	17.7778 %
59	mfcc_sma [14]	84.4444 %	15.5556 %

Tabla 11. Tabla fiabilidad/error bloque 4

BLOQUE 5

	CARACTERÍSTICA	CORRECTO	ERROR
1	audspec_lengthL1norm_sma_de_	84.4444 %	15.5556 %
2	audspecRasta_lengthL1norm_sma_de_	80 %	20 %
3	pcm_RMSenergy_sma_de_	75.5556 %	24.4444 %
4	pcm_zcr_sma_de_	71.1111 %	28.8889 %
5	audSpec_Rfilt_sma_de[0]	73.3333 %	26.6667 %
6	audSpec_Rfilt_sma_de [1]	62.2222 %	37.7778 %
7	audSpec_Rfilt_sma_de [2]	44.4444 %	55.5556 %
8	audSpec_Rfilt_sma_de [3]	71.1111 %	28.8889 %
9	audSpec_Rfilt_sma_de [4]	73.3333 %	26.6667 %
10	audSpec_Rfilt_sma_de [5]	75.5556 %	24.4444 %

ANEXO E: TABLA FIABILIDAD Y ERROR

11	audSpec_Rfilt_sma_de [6]	77.7778 %	22.2222 %
12	audSpec_Rfilt_sma_de [7]	71.1111 %	28.8889 %
13	audSpec_Rfilt_sma_de [8]	64.4444 %	35.5556 %
14	audSpec_Rfilt_sma_de [9]	77.7778 %	22.2222 %
15	audSpec_Rfilt_sma_de [10]	68.8889 %	31.1111 %
16	audSpec_Rfilt_sma_de [11]	71.1111 %	28.8889 %
17	audSpec_Rfilt_sma_de [12]	71.1111 %	28.8889 %
18	audSpec_Rfilt_sma_de [13]	62.2222 %	37.7778 %
19	audSpec_Rfilt_sma_de [14]	77.7778 %	22.2222 %
20	audSpec_Rfilt_sma_de [15]	77.7778 %	22.2222 %
21	audSpec_Rfilt_sma_de [16]	64.4444 %	35.5556 %
22	audSpec_Rfilt_sma_de [17]	77.7778 %	22.2222 %
23	audSpec_Rfilt_sma_de [18]	71.1111 %	28.8889 %
24	audSpec_Rfilt_sma_de [19]	73.3333 %	26.6667 %
25	audSpec_Rfilt_sma_de [20]	66.6667 %	33.3333 %
26	audSpec_Rfilt_sma_de [21]	75.5556 %	24.4444 %
27	audSpec_Rfilt_sma_de [22]	80 %	20 %
28	audSpec_Rfilt_sma_de [23]	75.5556 %	24.4444 %
29	audSpec_Rfilt_sma_de [24]	64.4444 %	35.5556 %
30	audSpec_Rfilt_sma_de [25]	66.6667 %	33.3333 %
31	pcm_fftMag_fband250-650_sma_de_	82.2222 %	17.7778 %
32	pcm_fftMag_fband1000-4000_sma_de_	77.7778 %	22.2222 %
33	pcm_fftMag_spectralRollOff25.0_sma_de_	73.3333 %	26.6667 %
34	pcm_fftMag_spectralRollOff50.0_sma_de_	82.2222 %	17.7778 %
35	pcm_fftMag_spectralRollOff75.0_sma_de_	68.8889 %	31.1111 %
36	pcm_fftMag_spectralRollOff90.0_sma_de_	64.4444 %	35.5556 %
37	pcm_fftMag_spectralFlux_sma_de_	84.4444 %	15.5556 %
38	pcm_fftMag_spectralCentroid_sma_de_	71.1111 %	28.8889 %
39	pcm_fftMag_spectralEntropy_sma_de_	80 %	20 %
40	pcm_fftMag_spectralVariance_sma_de_	80 %	20 %
41	pcm_fftMag_spectralSkewness_sma_de_	73.3333 %	26.6667 %
42	pcm_fftMag_spectralKurtosis_sma_de_	77.7778 %	22.2222 %
43	pcm_fftMag_spectralSlope_sma_de_	84.4444 %	15.5556 %
44	pcm_fftMag_psySharpness_sma_de_	66.6667 %	33.3333 %
45	pcm_fftMag_spectralHarmonicity_sma_de_	64.4444 %	35.5556 %
46	mfcc_sma_de[1]	55.5556 %	44.4444 %
47	mfcc_sma_de [2]	77.7778 %	22.2222 %
48	mfcc_sma_de [3]	75.5556 %	24.4444 %
49	mfcc_sma_de [4]	77.7778 %	22.2222 %
50	mfcc_sma_de [5]	75.5556 %	24.4444 %

ANEXO E: TABLA FIABILIDAD Y ERROR

51	mfcc_sma_de [6]	77.7778 %	22.2222 %
52	mfcc_sma_de [7]	73.3333 %	26.6667 %
53	mfcc_sma_de [8]	80 %	20 %
54	mfcc_sma_de [9]	75.5556 %	24.4444 %
55	mfcc_sma_de [10]	75.5556 %	24.4444 %
56	mfcc_sma_de [11]	75.5556 %	24.4444 %
57	mfcc_sma_de [12]	82.2222 %	17.7778 %
58	mfcc_sma_de [13]	75.5556 %	24.4444 %
59	mfcc_sma_de [14]	77.7778 %	22.2222 %

Tabla 12. Tabla fiabilidad/error bloque 5